

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Svit Timej Zebec

**iBeacon tehnologija pri razvoju  
mobilnih aplikacij**

DIPLOMSKO DELO  
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN MATEMATIKA

MENTOR: doc. dr. Dejan Lavbič

Ljubljana 2015



Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Področje interneta stvari v zadnjem času pridobiva na pomembnosti, saj omogoča povezovanje različnih naprav in izboljšanje delovnih procesov na različnih področjih. Eno takšnih področjih je zagotovo zdravstvo, kjer se kažejo številne možnosti izboljšanja poslovanja, predvsem pri uvedbi brezpapirnega poslovanja. Ena izmed novejših tehnologij, povezana z omenjeno problematiko, je iBeacon, ki za delovanje uporablja nizko-energijski bluetooth in nam omogoča razvoj mobilnih aplikacij na podlagi bližine. Študent naj v okviru diplomske naloge kritično predstavi uporabno vrednost tehnologije iBeacon pri razvoju mobilnih aplikacij. Na podlagi pregledane znanstvene in strokovne literature naj identificira različne primere uporabe in izzive pri tej uporabi. Za potrditev svojih ugotovitev naj razvije prototip aplikacije na področju zdravstva za obe najbolj pogosto uporabljani platformi Android in iOS ter jih oceni s SWOT analizo.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Svit Timej Zebec sem avtor diplomskega dela z naslovom:

*iBeacon tehnologija pri razvoju mobilnih aplikacij*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Dejana Lavbiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 7. septembra 2015

Podpis avtorja:





*Zahvaljujem se svojemu mentorju, doc. dr. Dejanu Lavbiču, za vso pomoč in usmerjanje pri izdelavi diplomskega dela ter za izjemno hitre odgovore na moja vprašanja v trenutkih, ko sem jih najbolj potreboval.*

*Hvala podjetju Marand in Alešu Smokvini za pomoč pri izboru teme diplomskega dela, za vse izposojene iBeacone ter za pomoč pri razvoju iBeacon aplikacij.*

*Zahvaljujem se staršema za podporo in spodbudo skozi vsa ta leta, za vse nasvete v težkih trenutkih ter za prevzem nehvaležne vloge lektorja diplomske naloge.*

*Hvala mojim IŠRM kolegom, brez katerih opravljanje dodiplomskega študija ne bi bilo pol tako uspešno in zabavno.*

*Hvala Tjaša, za pomoč pri lektoriranju in za vlivanje poguma in upanja v trenutkih, ko to najbolj potrebujem.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>iBeacon tehnologija</b>	<b>3</b>
2.1	Bluetooth BLE . . . . .	3
2.1.1	Model BLE protokola . . . . .	4
2.2	iBeacon standard . . . . .	7
2.2.1	Oglaševalni podatki iBeacona . . . . .	8
2.2.2	Oglaševalni interval . . . . .	9
2.3	Alternativni standardi . . . . .	10
2.4	Lokacija in bližina . . . . .	11
2.5	iBeacon funkcije . . . . .	13
2.5.1	iBeacon regija . . . . .	13
2.5.2	Spremljanje . . . . .	13
2.5.3	Odkrivanje . . . . .	14
2.6	Konfiguracija iBeaconov . . . . .	17
<b>3</b>	<b>iBeaconi pri izdelavi mobilnih aplikacij</b>	<b>21</b>
3.1	Internet stvari (IoT) . . . . .	21
3.2	Pomembni primeri uporabe iBeaconov . . . . .	22
3.3	Primeri in analiza obstoječih aplikacij z iBeacon . . . . .	24

## KAZALO

3.3.1	Launch here . . . . .	24
3.3.2	Aplikacija za pomoč slabovidnim uporabnikom pri navigaciji po letališču . . . . .	25
3.3.3	Primerjava med aplikacijama . . . . .	26
3.4	Postavitev velikega števila iBeaconov . . . . .	27
3.4.1	Postavitev in vzdrževanje . . . . .	27
3.4.2	Uporaba v zdravstvu . . . . .	29
<b>4</b>	<b>Razvoj aplikacije za prikaz podatkov pacienta na platformah Android in iOS</b>	<b>31</b>
4.1	Opis in cilj razvitih aplikacij . . . . .	31
4.1.1	Opis . . . . .	31
4.1.2	Cilji . . . . .	31
4.2	Uporabljene tehnologije, razvojna orodja in knjižnice . . . . .	32
4.2.1	Android aplikacija . . . . .	32
4.2.2	iOS aplikacija . . . . .	34
4.2.3	Obe aplikaciji . . . . .	35
4.2.4	Strežnik . . . . .	35
4.3	Tehničen opis delovanja aplikacij . . . . .	36
4.3.1	Delovanje glavnih funkcionalnosti aplikacij . . . . .	36
4.3.2	Delovanje grafičnega vmesnika aplikacij . . . . .	41
4.4	Primerjava razvoja iBeacon aplikacij na platformah Android in iOS	44
4.5	SWOT analiza . . . . .	45
4.5.1	Prednosti . . . . .	45
4.5.2	Slabosti . . . . .	46
4.5.3	Priložnosti . . . . .	46
4.5.4	Grožnje . . . . .	47
<b>5</b>	<b>Zaključek</b>	<b>49</b>

# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>BLE</b>	Bluetooth Low Energy	Nizko-energijski bluetooth
<b>IoT</b>	Internet of Things	Internet stvari
<b>UUID</b>	Universal Unique Identifier	Univerzalni enolični identifikator
<b>SDK</b>	Software development kit	Orodja za razvoj programske opreme
<b>RSSI</b>	Received Signal Strength Indication	Indikacija moči prejetega signala
<b>LE</b>	Low Energy	Nizko-energijski
<b>IDE</b>	Integrated development environment	Integrirano razvojno okolje
<b>GPS</b>	Global Positioning System	Globalni sistem pozicioniranja
<b>API</b>	Aplication programming interface	Aplikacijski programski vmesnik
<b>HTTP</b>	Hypertext Transfer Protocol	Protokol za izmenjavo hiperteksta
<b>SWOT</b>	Strengths, weaknesses, opportunities and threats	Prednosti, slabosti, priložnosti in grožnje



# Povzetek

iBeacon tehnologija, ki za delovanje uporablja BLE, omogoča, da načrtujemo mobilne aplikacije na osnovi bližine. Zanima nas, kako lahko koncept bližine integriramo v razvoj mobilnih aplikacij. V diplomskem delu si najprej podrobneje ogledamo iBeacon tehnologijo, njeni funkciji spremljanje in odkrivanje, koncepta bližine in lokacije ter konfiguracijo iBeaconov. Nato analiziramo dve obstoječi iBeacon aplikaciji, pregledamo različne primere uporabe in jih razdelimo v tri širše skupine. Predstavimo možnosti postavitve in vzdrževanja velikega števila iBeaconov. Na koncu želimo podati primer Android in iOS mobilne aplikacije, ki uporablja iBeacon tehnologijo za prikazovanje podatkov o pacientu. Na podlagi razvoja aplikacij primerjamo razvoj iBeacon mobilnih aplikacij glede na platformo. Na koncu še analiziramo razviti aplikaciji z uporabo SWOT analize.

**Ključne besede:** BLE, iBeacon, bližina, mobilna aplikacija.





# Abstract

iBeacon technology, made possible by BLE, enables mobile application developers to base their applications on proximity. We would like to know how proximity can be integrated in the development of mobile applications. In the thesis we take a deeper look at iBeacon technology where monitoring and ranging are examined and the differences between location and proximity explained. Available options used for the configuration of iBeacons are also presented. Then two existing iBeacon applications are analyzed. Following that different use cases are listed and sorted into three groups. Large scale iBeacon deployment options are presented and examined. Lastly an Android and an iOS application which use iBeacon technology for displaying patient data are developed. A comparison between the development of iBeacon applications on both platforms is made based on the development of the applications. At the end the developed application are analyzed with the use of a SWOT analysis.

**Keywords:** BLE, iBeacon, proximity, mobile application.



# Poglavje 1

## Uvod

Razvoj in vedno večja razširjenost uporabe pametnih mobilnih naprav prinašata potrebo po razvoju kvalitetnih in uporabnih mobilnih aplikacij za osebno ali profesionalno uporabo. Mobilne aplikacije lahko z uporabo iBeacon tehnologije implementirajo funkcionalnosti, ki temeljijo na bližini med mobilno in iBeacon napravo oziroma iBeaconom. Z uporabo koncepta bližine med napravo in objektom ali območjem (ki ga predstavlja iBeacon), lahko mobilna aplikacija veliko problemov reši na načine, ki so preprostejši in bolj intuitivni za uporabnika. V primeru da je razvijalec aplikacije soočen s problemom, za katerega misli, da bi ga lahko bolje rešil z uporabo iBeacon tehnologije, si lahko zastavi nekaj vprašanj: kaj natančneje je iBeacon tehnologija in kakšne funkcionalnosti ponuja, kakšne so omejitve razvoja iBeacon aplikacije na Android in kakšne na iOS platformi, kakšni so konkretni primeri uporabe iBeaconov in kako se lotimo postavitve in konfiguracije iBeaconov. Cilj diplomskega dela je formalen odgovor na ta vprašanja in razvoj vzorčne aplikacije, ki uporablja iBeacon tehnologijo.

V drugem poglavju diplomske naloge je podrobneje predstavljena iBeacon tehnologija, v tretjem poglavju je raziskan pomen iBeacon tehnologije pri izdelavi mobilnih aplikacij, četrto poglavje pa vključuje predstavitev in analizo lastno razvite Android in iOS iBeacon aplikacije.

Odgovori na prej zastavljena vprašanja so osnovani na podlagi teoretične in empirične raziskave. Teoretični del obsega drugo in tretje poglavje diplomske naloge,

empirični del pa četrto poglavje. Na osnovi izkušenj, pridobljenih pri razvoju aplikacij, so predstavljene prednosti in slabosti razvoja iBeacon aplikacij ter razlike pri uporabi iBeacon tehnologije na obeh platformah. Razviti aplikaciji predstavljata vzorčni primer uporabe iBeacon tehnologije v praksi in služita kot osnova za razvoj kompleksnejših aplikacij in rešitev, namenjenih za uporabo v bolnišničnih okoljih.

## Poglavje 2

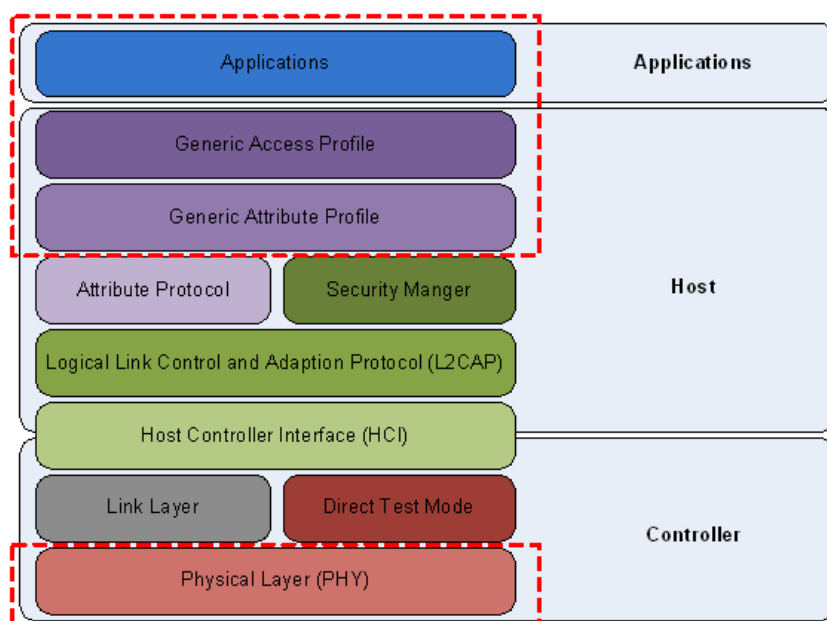
# iBeacon tehnologija

### 2.1 Bluetooth BLE

Nizko-energijski bluetooth (angl. bluetooth low energy) je nova brezžična tehnologija, razvita s strani Bluetooth Special Interest Group (SIG), ki se uporablja za komunikacijo kratkega dosega. BLE je lastnost Bluetooth 4.0 specifikacije in ni kompatibilna s starejšimi Bluetooth specifikacijami. Bluetooth 4.0 specifikacija dopušča, da naprave implementirajo le BLE, le klasičen bluetooth, ali obe tehnologiji hkrati. Glavni prednosti BLE, v primerjavi s klasičnim bluetoothom, sta veliko nižja poraba energije in cena. Kljub temu BLE ohranja podoben domet kot klasičen bluetooth. Prednost BLE pred alternativnimi brezžičnimi nizko-energijskimi tehnologijami, kot so ZigBee, 6LoWPAN ali Z-Wave je v tem, da BLE predstavlja eno-skočno (angl. single-hop) rešitev, saj vsi ostali omenjeni potrebujejo več-skočno (angl. multi-hop) mreženje. Posledično je BLE veliko bolj uporaben na področju zdravstva, potrošniške elektronike, varnosti, športa itd. Poleg tega je bluetooth tehnologija zelo razširjena med mobilnimi telefoni, tablicami, prenosnimi računalniki in ostalimi mobilnimi napravami. Splošna razširjenost klasičnega bluetootha med omenjenimi napravami daje BLE tehnologiji prednost pri širjenju uporabe, saj ima implementacija BLE veliko podobnosti implementaciji klasičnega bluetootha.

### 2.1.1 Model BLE protokola

Model BLE protokola je kot pri klasičnem bluetoothu sestavljen iz dveh delov, in sicer iz krmilnika (angl. controller) in gostitelja (angl. host). Krmilnik je sestavljen iz fizične (angl. physical layer) in linijske (angl. link layer) ravni in je ponavadi implementiran kot majhen sistem na čipu (angl. system on a chip) z integriranim radiem. Gostitelj teče na aplikacijskem procesorju in vsebuje višje-nivojske plasti. Te plasti so L2CAP – Logical Link Control and Adaptation Protocol, ATT – Attribute Protocol, GATT – Generic Attribute Profile, SMP – Security Manager Protocol in GAP – Generic Access Profile [12].



Slika 2.1: Model BLE protokola [7]

#### Fizični nivo

BLE operira v 2.4 GHz ISM<sup>1</sup> frekvenčnem pasu in zajema 40 radijskih frekvenčnih kanalov, ki se delijo v dva tipa, in sicer na oglaševalne kanale ter podatkovne

<sup>1</sup>Industrijski, znanstveni in medicinski radijski pasovi (angl. industrial, scientific and medical radio bands) so deli radijskega spektruma, ki so mednarodno rezervirani za uporabo v industrijske, znanstvene in medicinske namene.

kanale. Oglaševalni kanali so trije, podatkovnih kanalov pa je 37. Oglaševalni kanali služijo za odkrivanje naprav, vzpostavitev povezave ter prenos oddaje, medtem ko podatkovni kanali služijo za dvosmerno komunikacijo med povezanimi napravama.

### **Linijski nivo**

V primeru, da želi naprava z BLE le oddajati podatke, slednja prenaša podatke v obliki oglaševalnih paketkov preko oglaševalnih kanalov. V tem primeru se naprava imenuje oglaševalec (angl. advertiser). Za dvosmerno komunikacijo med napravama je potrebna povezava med njima. Postopek povezave med napravama je sledeč: oglaševalec objavi po oglaševalskih kanalih, da je naprava povezljiva, medtem ko druga naprava posluša in skuša odkriti povezljive naprave. Ko naprava, ki posluša, odkrije ustreznega povezljivega oglaševalca, se poskuša z njim povezati, tako da mu pošlje sporočilo za zahtevo povezave, ki vzpostavi točkovno povezavo med napravama. Naprava, ki je sprožila povezavo, zavzame vlogo gospodarja (angl. master), oglaševalec pa zavzame vlogo sužnja (angl. slave). Gospodar je lahko naenkrat povezan z več sužnji, medtem ko je suženj lahko naenkrat povezan le z enim gospodarjem. Tako več BLE naprav tvori omrežje, ki je sestavljeno iz enega gospodarja in več sužnjev, ter se imenuje piko-omrežje (angl. piconet).

### **L2CAP – Logical Link Control and Adaptation Protocol**

L2CAP v BLE predstavlja optimiziran in poenostavljen protokol, ki temelji na L2CAP protokolu iz klasičnega bluetootha. Glavna funkcija L2CAP je multipleksiranje podatkov med različnimi višjenivojskimi protokoli.

### **ATT – Attribute Protocol**

ATT definira komunikacijo med dvema napravama, ki zavzameta vlogi strežnika in odjemalca, nad L2CAP kanalom. Vlogo strežnika ali odjemalca definira GATT. Ti dve vlogi sta neodvisni od vlog gospodarja in sužnja. Odjemalec lahko dostopa

do atributov strežnika s pošiljanjem zahtevkov, na katere se strežnik odziva z odgovorom.

### **GATT – Generic Attribute Profile**

GATT definira ogrodje, ki uporablja ATT za odkrivanje storitev in izmenjavo karakteristik med dvema napravama.

### **SMP – Security Manager Protocol**

SMP omogoča dva načina varovanja, in sicer LE varovalni način 1 (angl. LE Security Mode 1) in LE varovalni način 2 (angl. LE Security Mode 2). Oba načina se delita na več ravni [24]. Ravni prvega načina uporabljajo šifriranje podatkov, ravni drugega pa podpisovanje podatkov.

Ravni LE varovalnega načina 1:

- raven 1 ne specificira nobene varnosti, torej ni avtentikacije in šifriranja,
- raven 2 zahteva neavtentificirano parjenje z šifriranjem,
- raven 3 zahteva avtentificirano parjenje z šifriranjem.

Ravni LE varovalnega načina 2:

- raven 1 zahteva neavtentificirano parjenje s podpisovanjem podatkov,
- raven 2 zahteva avtentificirano parjenje s podpisovanjem podatkov.

### **GAP – Generic Access Profile**

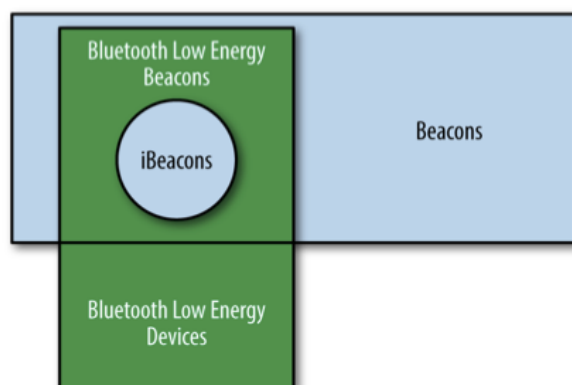
GAP je najvišji nivo modela BLE protokola, ter določi vloge naprav, načine in procedure za odrivanje naprav in storitev, upravljanje vzpostavitve povezave in varnost. Vloge naprav, ki jih specificira GAP, so oddajnik (angl. broadcaster), opazovalec (angl. observer), periferni (angl. peripheral) in centralni (angl. central). Naprava z vlogo oddajnika lahko le oddaja podatke in se ne more povezovati



z drugimi napravami. Vloga opazovalca je namenjena sprejemanju podatkov, ki jih oddaja oddajnik. Vloga centralnega je primerna za napravo, ki vzpostavlja in upravlja več povezav hkrati, medtem ko naprava z vlogo periferne lahko vzpostavi le eno povezavo z napravo z vlogo centralnega.

## 2.2 iBeacon standard

iBeacon tehnologijo je razvil Apple in je bila predstavljena na Svetovni razvijalski konferenci 2013 (angl. WorldWide Developers Conference 2013 – WWDC 2013). iBeacon tehnologija omoča, da mobilne naprave na platformi iOS poslušajo signale, ki jih oddajajo t. i. beaconi. Uporaba iBeacon tehnologije na platformi Android je možna z uporabo SDK-jev ali knjižnic, ki jih ponujajo proizvajalci beaconov oziroma kakšna druga skupina. Beaconi so aktivne naprave, običajno z lastnim napajanjem, ki z uporabo BLE tehnologije periodično oddajajo podatke o sebi, katere mobilna naprava, ki uporablja iBeacon tehnologijo, zazna in preko mobilnih aplikacij preslika v akcije [9]. Beaconi, ki uporabljajo iBeacon protokol, so imenovani iBeaconi. Relacijo med BLE, beaconi in iBeaconi vidimo na sliki 2.2. Kot posledica tega, da so beaconi v kontekstu BLE le oddajniki (angl. broadcaster), je iBeacon protokol zelo preprost in predstavlja le način prenašanja nekaj številskih identifikatorjev med beaconom in mobilno napravo. Preslikava teh preprostih informacij v kompleksne akcije je naloga mobilne aplikacije, kar predstavlja največji izziv za razvijalce aplikacij.



Slika 2.2: Relacija med BLE, beaconi in iBeaconi [11]

iBeacon je torej naprava, ki uporablja iBeacon protokol za periodično oddajanje informacij o sebi. Tipično so iBeaconi naprave, ki imajo nameščen BLE modul, namenjene izključno oddajanju informacij o sebi. Kljub temu so lahko iBeaconi tudi kompleksnejše naprave, kot so iOS in Android mobilne naprave, ali preprostejše naprave, kot sta Raspberry pi<sup>2</sup> in LightBlue Bean<sup>3</sup>. Oddajanje podatkov iBeacona poteka tako, da iBeacon preko BLE modula periodično oddaja oglaševalne pakete (angl. advertising packets), ki vsebujejo podatke o napravi. Oglaševalni paketi so zasnovani za to, da so posredovani prejemnikom, se pa na njih ni potrebno odzvati. To pomeni, da se povezava med napravami ne vzpostavi, ampak prejemnik le prejme oglaševalne pakete, ter se primerno (ne)odzove.

### 2.2.1 Oglaševalni podatki iBeacona

Oglaševanje iBeacona si lahko predstavljamo kot pozdrav iBeacona svetu, v katerem pravi: "Zdravo, jaz sem tukaj in moje ime je ...". Ime iBeacona je sestavljeno iz Univerzalnega enoličnega identifikatorja (angl. Universal Unique Identifier – UUID), glavnega (angl. major) števila in drugotnega števila (angl. minor).

- **UUID** je 128 bitni identifikator, ki praviloma enolično določi organizacijo,

<sup>2</sup>Majhen računalnik, zgrajen le na eni plošči z vezjem, namenjen spodbujanju učenja osnovnih računalniških znanj v šolah.

<sup>3</sup>BLE Arduino mikro-krmilnik, ki omogoča brezžično programiranje.

ki ji iBeacon pripada. Čeprav UUID v imenu nosi informacijo o tem, da je enoličen identifikator, v resnici ni zares enoličen, saj ga lahko spremenimo na poljubno vrednost;

- **glavno število** je 16 bitno število, za katerega bluetooth in iBeacon specifikaciji ne definirata, kako naj bi se uporabljal. Kljub temu se ponavadi glavno število uporablja za identifikacijo skupine večih iBeaconov v okviru ene organizacije. Na primeru verige trgovin bi to pomenilo, da če UUID označuje organizacijo, kateri pripada iBeacon, glavno število označuje vse beacons v posamezni trgovini;
- **drugotno število** je 16 bitno število, čigar uporaba, kot pri glavnemu številu, ni striktno definirana, ampak se ponavadi uporablja za identifikacijo posameznih iBeaconov. Če se vrnemo na primer verige trgovin, bi drugotno število označevalo posamezen iBeacon v trgovini.

Poleg teh številskih identifikatorjev iBeacon oglašuje tudi parameter, imenovan **izmerjena moč** (angl. *measured power*). Ta parameter je določen ob kalibraciji iBeacon naprave s strani proizvajalca. Izmerjena moč označuje povprečno moč zaznanega signala, ki prihaja iz iBeacona, na referenčni napravi, pri razdalji 1 m med napravo in iBeaconom. Vrednost tega parametra se uporablja pri izračunavanju razdalje med napravo in iBeaconom. Postopek izračunavanja razdalje do iBeacona je podrobneje predstavljen v razdelku 2.5.3.

Konfiguracija oglaševalnih podatkov iBeacona bo predstavljena v razdelku 2.6.

### 2.2.2 Oglaševalni interval

Oglaševalni interval (angl. *advertising interval*) določa s kakšno časovno frekvenco iBeacon oglašuje svoje podatke. Bluetooth specifikacija dopušča precejšno svobodo glede nastavitve oglaševalnega intervala, medtem ko je iBeacon specifikacija zelo striktna in fiksira interval oddajanja na 100 ms. Kljub temu, da iBeacon specifikacija fiksira interval oddajanja, ga lahko na nekaterih iBeaconih vseeno nastavljamo, kar je odvisno od proizvajalca iBeacona. Oglaševalni interval je zelo pomembna nastavev iBeacona, saj lahko premajhen interval povzroči

preveliko porabo energije in s tem kratko avtonomijo baterije. Prevelik interval pa za nekatere storitve, ki uporabljajo iBeacone, ni dovolj dober. Bolj podroben vpogled v nastavitve oglaševalnega intervala sledi v razdelku 2.6.

## 2.3 Alternativni standardi

Poleg iBeacon tehnologije obstaja še nekaj podobnih tehnologij, ki uporabljajo BLE za periodično oglaševanje svojih podatkov in jih lahko uvrstimo pod skupno ime beacon tehnologije. Glavni alternativni iBeacon tehnologiji sta Eddystone [16] in AltBeacon [6].

**Eddystone** je nova<sup>4</sup> beacon tehnologija razvita s strani Google. Oglaševalni paketi, ki jih Eddystone oddaja so:

- Eddystone-UID – enolični identifikator beacons, podobno kot UUID iBeacona oziroma kombinacija UUID, glavnega števila in drugotnega števila;
- Eddystone-URL – URL, ki ga beacon oddaja in ga lahko mobilna naprava sprejme ter odpre v brskalniku;
- Eddystone-TLM – telemetrija oziroma podatki pridobljeni s senzorji.

Eddystone specifikacija torej v primerjavi z iBeacon specifikacijo omogoča oddajanje URL naslova, ter telemetrije. Pomembno je zlasti oglaševanje URL naslova, saj to pomeni, da lahko beacon služi kot preusmeritev na spletno stran oziroma spletno aplikacijo. V tem primeru uporabniku ni potrebno prenašati posebne aplikacije za interakcijo z beaconom, saj za odpiranje URL-ja zadostuje spletni brskalnik. Poleg tega je Eddystone specifikacija, v nasprotju z iBeacon specifikacijo, odprtokodna, kar omogoča večjo transparentnost in boljšo integracijo v razvijalsko skupnost. Eddystone prav tako že v osnovi podpira iOS in Android platformo, medtem ko iBeacon nudi podporo za iOS, ter se pri Androidu zanaša

---

<sup>4</sup>Eddystone je Google predstavil julija, 2015.

na tujo podporo, v obliki SDK-jev in knjižnic. Eddystone je zagotovo zelo obetavna alternativa iBeacon tehnologiji zaradi dodatnih možnosti, ki jih ponuja in ker ima za sabo podporo podjetja Google.

**AltBeacon** je odprtokodna beacon tehnologija podjetja Radius Networks. V primerjavi z Eddystone je AltBeacon tehnologija veliko bolj podobna iBeacon tehnologiji, saj omogoča iste funkcionalnosti kot iBeacon tehnologija. AltBeacon lahko prilagodimo, da se obnaša kot iBeacon<sup>5</sup>, saj je oglaševalni paket AltBeacona sestavljen tudi iz 20 bajtov velikega "Beacon ID", katerega lahko spremenimo v iBeacon format, in sicer 16 bajtov velik UUID, 2 bajta veliko glavno število in 2 bajta veliko drugotno število. Poleg tega lahko del beacon ID-ja, ki predstavlja UUID, uporabimo drugače kot pri iBeaconu, kjer ponavadi predstavlja identifikator organizacije, saj lahko iBeacon podatke o organizaciji hrani v nastavljenem številu velikost 2 bajta. V splošnem AltBeacon omogoča več prilagodljivosti pri nastavitvi svojega oglaševalnega paketa, s čimer lahko pokrije več primerov uporabe, vendar lahko s prilagajanjem oglaševalnega paketa implementacija postane veliko bolj zahtevna. Glavni cilji AltBeacon specifikacije so transparentnost, prilagodljivost in podpora za naprave in platforme, za katere iBeacon tehnologija ne ponuja podpore. Za uporabo iBeacon tehnologije na Android platformi se večinoma uporablja AltBeacon knjižnica, ki je v osnovi prilagojena za AltBeacon specifikacijo, vendar jo lahko s spremembo ene izmed nastavitev uporabljamo tudi za odkrivanje iBeaconov.

## 2.4 Lokacija in bližina

Lokacija (angl. location) in bližina (angl. proximity) sta pojma, ki imata podoben pomen, saj oba nosita informacijo o tem, kje se nekaj nahaja. Razlika med njima je v tem, da je lokacija absolutna, bližina pa relativna v odnosu do nečesa drugega, ponavadi objekta. Če poskusimo lokacijo in bližino kvantificirati, je

---

<sup>5</sup>V obratno smer ta posplošitev ne deluje, torej iBeacona ne moremo prilagoditi, da se obnaša kot AltBeacon.

lokacija vektorska količina, bližina pa skalarna. To pomeni, da bližina nima smeri in je predstavljena le s skalarjem. Primerjava med obema pojmom je prikazana v tabeli 2.1. Ko razmišljamo o lokaciji, si lahko zastavimo vprašanje: “Kaj je moja lokacija?” Na to vprašanje lahko odgovorimo s pomočjo tehnologij kot sta GPS<sup>6</sup> in WiFi<sup>7</sup>. Odgovor na to vprašanje je lahko v obliki geografskih koordinat, naslova, ulice, zgradbe ... Kadar razmišljamo o bližini, je bolj ustrezno vprašanje, ki si ga lahko zastavimo: “Kaj je blizu mene?” Vprašanje lahko tudi glede na kontekst obrnemo: “Blizu česa sem jaz?” Na takšni vprašanji lahko poskusimo odgovoriti podobno kot na prejšnje, torej z GPS ali WiFi, vendar lahko odgovor v veliko situacijah dobimo lažje in učinkoviteje s pomočjo tehnologije kot je iBeacon. Slednja lahko oceni razdaljo med uporabnikom (mobilno napravo) in objektom, ki je predstavljen z iBeaconom. Načini, kako iBeacon tehnologija določa bližino med uporabnikom oziroma napravo in iBeaconom, so predstavljeni v razdelku 2.5.

Lokacija	Bližina
GPS, Wifi	iBeacon, BLE
globalna navigacija	pozicija v notranjosti
več sto metrov, kilometri	centimetri, metri

Tabela 2.1: Primerjava med lokacijo in bližino

Storitve, ki temeljijo na bližini, ugotavljajo kdaj je uporabnik blizu enemu ali več objektom, ter te informacije spreminjajo v ustrezne akcije. Primeri so predstavljeni v razdelku 3.2. Lahko pa poskušajo na podlagi fiksnih točk in razdalje uporabnika do njih določati mikrolokacijo uporabnika. Z uporabo iBeacon tehnologije je takšna storitev implementirana tako, da iBeaconi predstavljajo fiksne točke v prostoru in se pozicija uporabnika izračuna na podlagi razdalje do iBe-

<sup>6</sup>Globalni sistem pozicioniranja (angl. Global Positioning System) je navigacijski sistem, ki za določanje lokacije uporablja satelite v vesolju.

<sup>7</sup>Če je naprava povezana v WiFi omrežje, lahko določimo okvirno lokacijo naprave, saj je lokacija WiFi omrežja znana.

aconov, ki so v dometu. Določanje mikrolokacije je eden izmed kompleksnejših primerov uporabe iBeaconov. Več informacij o tem primeru uporabe sledi v razdelku 3.2.

## 2.5 iBeacon funkcije

iBeacon tehnologija ponuja za odkrivanje iBeaconov dve različni funkciji, in sicer spremljanje (angl. monitoring) in odkrivanje (angl. ranging). Ko govorimo o spremljanju in odkrivanju moramo omeniti, kaj je iBeacon regija, saj je pomembna pri obeh funkcijah, zlasti pri spremljanju.

### 2.5.1 iBeacon regija

iBeacon regija je določena z enim ali več iBeaconi. Je unija območij vsebujočih iBeaconov, pri čemer je posamezno območje omejeno z dometom pripadajočega iBeacona. Regija je lahko določena na tri načine:

- samo UUID: regijo določajo vsi iBeaconi, ki ustrezajo UUID številu regije, ne glede na vrednosti glavnega in drugega števila;
- UUID in glavno število: regijo določajo vsi iBeaconi, ki ustrezajo UUID in glavnemu številu regije, ne glede na vrednost drugega števila;
- UUID, glavno in drugotno število: regijo določajo vsi iBeaconi, ki ustrezajo UUID, glavnemu in drugotnemu številu regije. Glede na to, da je drugotno število identifikator na najnižjem nivoju (označeval naj bi posamezen iBeacon), je ponavadi regija ob taki določitvi, regija enega iBeacona.

### 2.5.2 Spremljanje

Spremljanje določa ali je mobilna naprava znotraj ali izven poljubne iBeacon regije. Spremljanje deluje na iBeacon regiji in ima le dve stanji: naprava se nahaja v regiji in naprava se nahaja izven regije. Ker je spremljanje vezano na posamezno regijo, se pogosto uporablja izraz spremljanje regije (angl. region monitoring),

za regijo na kateri se izvaja spremljanje pa spremljana regija. Mobilna naprava lahko spremljanje izvaja na večih regijah hkrati, vendar je število teh omejeno. Na platformi iOS je omejeno na največ 20 sočasno spremljanih regij.

Pri spremljanju sta ključna dva dogodka, in sicer vstop naprave v spremljano regijo in izstop iz spremljane regije. Mobilna naprava začne prejemati dogodke o vstopu oziroma izstopu iz regije po tem, ko začne spremljanje regije. Dogodek o vstopu v regijo prejme, ko zazna kakšnega izmed iBeaconov, s katerimi je regija določena. Dogodek o izstopu iz regije pa prejme, ko preneha zaznavati iBeacone, ki določujejo regijo. Ob prejetju dogodka se nato naprava ustrezno odzove. Spremljanje je namenjeno uporabi, ko je mobilna aplikacija v ospredju in v ozadju. V praksi na platformi iOS to pomeni, da tudi ko aplikacija deluje v ozadju, prejme dogodke o vstopu/izstopu iz regije, ter nato za nekaj sekund preide v ospredje, da se lahko ustrezno odzove na dogodek<sup>8</sup>. Na platformi Android je obnašanje spremljanja v ozadju odvisno od SDK-ja ali knjižnice, saj iBeacon tehnologija ne nudi podpore za Android. Spremljanje ne porabi veliko energije, kar je tudi eden izmed razlogov, da je primeren za uporabo, ko aplikacija teče v ozadju.

Zaradi nezanesljivosti brezžičnega signala ponavadi operacijski sistemi, ki skrbijo za sprožitev dogodkov o vstopu oziroma izstopu iz regije, implementirajo histerozo<sup>9</sup> za glajenje sporočanja dogodkov. V primeru, da ne bi bil implementiran noben mehanizem za robustnost sporočanja dogodkov, bi lahko naprava prejela dogodke o vstopu ali izstopu iz regije, tudi kadar se v resnici vstop ali izstop iz regije ne bi zgodil.

### 2.5.3 Odkrivanje

Odkrivanje omogoča odkritje vseh iBeaconov, ki so vidni mobilni napravi in omogoča izračun dolžine med napravo in posameznim iBeaconom. Odkrivanje deluje

---

<sup>8</sup>To je seveda poenostavljena razlaga prehajanja med aplikacijskimi stanji v sistemu iOS, vendar služi osnovni ilustraciji delovanja spremljanja v ozadju.

<sup>9</sup>Značilnost sistemov, da se na sile ne odzovejo takoj, temveč počasi, ali da se ne povrnejo popolnoma v svoje prvotno stanje: to je, sistemi, katerih stanja so odvisna od njihove neposredne preteklosti [20].



tako, da naprava periodično prejema seznam vseh vidnih iBeaconov. Signal posameznega iBeacona, ki ga prejme naprava, ima izmerljivo moč, ki se imenuje RSSI – indikator moči prejetega signala (angl. received signal strength indicator). Na podlagi RSSI in izmerjene moči, predstavljene v razdelku 2.2.1, lahko naprava izračuna približno dolžino med sabo in iBeaconom. Natančnost izračunane dolžine je odvisna od natančnosti izmerjene moči (predstavljena v razdelku 2.2.1). Dokler naprava izvaja odkrivanje, dobiva periodično seznam vseh vidnih iBeaconov ter ocenjene dolžine vseh iBeaconov do naprave. Na iOS napravah meri časovni interval pridobivanja podatkov o vidnih iBeaconih približno 1 s in ga ni mogoče spreminjati. Na Android napravah je interval zopet odvisen od SDK-ja oziroma knjižnice. Ponavadi je v osnovi, podobno kot pri iOS napravah, nastavljen na približno 1 s, vendar nekateri SDK-ji oziroma knjižnice omogočajo spreminjanje intervala na Android napravah.

Odkrivanje je energetsko precej potratno, kar je eden izmed razlogov, da je namenjeno uporabi, ko aplikacija teče v ospredju. Na iOS platformi je sicer mogoče pripraviti aplikacijo, da izvaja odkrivanje v ozadju (omejeno časa), vendar je to v nasprotju z Applovimi smernicami, ki svetujejo uporabo odkrivanja le v ospredju. Drugi pomemben razlog za omejitev uporabe odkrivanja na uporabo, ko aplikacija teče v ospredju, je zasebnost uporabnika. Če se izvaja odkrivanje v ozadju, je lahko funkcionalnost odkrivanja izkoriščena za spremljanje gibanja uporabnika.

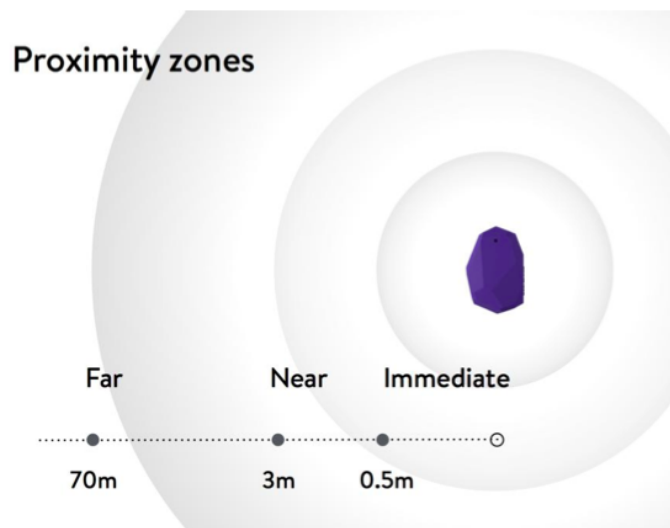
Poleg ocenjene razdalje med napravo in iBeaconom, obstaja še en način določanja kako daleč je iBeacon od naprave, in sicer z bližinskimi stanji (angl. proximity states). Vsak viden iBeacon ima med izvajanjem odkrivanja podatek v katerem bližinskem stanju se nahaja. Definirana so štiri bližinska stanja:

- neposredna bližina (angl. immediate) – kadar je iBeacon v neposredni bližini naprave, kar je manj kot 0.5 metra;
- blizu (angl. near) – kadar je iBeacon blizu naprave, kar je med 0.5 in 3 metra;
- daleč (angl. far) – kadar je iBeacon daleč od naprave, kar je vse od 3 do

70 metrov. Razpon tega intervala je odvisen od dometa signala iBeacona. Zgornja meja ni vedno 70 metrov, saj je to le ocena za domet iBeacona ob standardni nastavitvi moči oddajanja;

- neznano (angl. unknown) – bližina iBeacona ne more biti določena, najverjetneje, ker se je odkrivanje pravkar začelo oziroma je na voljo premalo meritev za določitev bližinskega stanja.

Grafično predstavitev bližinskih stanj lahko vidimo na sliki 2.3. Odkrivanje je primerno za implementiranje funkcionalnosti, kot so: odkrivanje kateri iBeacon je uporabniku najbližje, notranje lociranje oziroma določanje mikrolokacije, prikazovanje vseh vidnih iBeaconov. Iz izkušenj z uporabo odkrivanja je določanje razdalje med napravo in iBeaconom na trenutke precej nezanesljivo, zato je za zahtevnejše storitve, kot je določanje mikrolokacije, priporočljiva uporaba številnih, gosto postavljenih iBeaconov.



Slika 2.3: iBeacon regija z bližinskimi stanji [10]

## 2.6 Konfiguracija iBeaconov

Pri konfiguraciji iBeaconov iščemo ugodno razmerje med življenjsko dobo baterije iBeacona in še vedno primernimi nastavitvami glede na potrebe mobilne aplikacije, ki bo uporabljala iBeacone. Nastavljiva parametra iBeacona sta moč oddajanja (angl. *broadcasting power*) in interval oddajanja. Večja kot bo moč oddajanja in manjši kot bo interval oddajanja, krajša bo življenjska doba baterije. Izziv konfiguracije iBeacona je ugotoviti, do kakšne mere lahko znižamo moč oddajanja oziroma povečamo interval oddajanja, da ohranimo pričakovano funkcionalnost mobilne aplikacije.

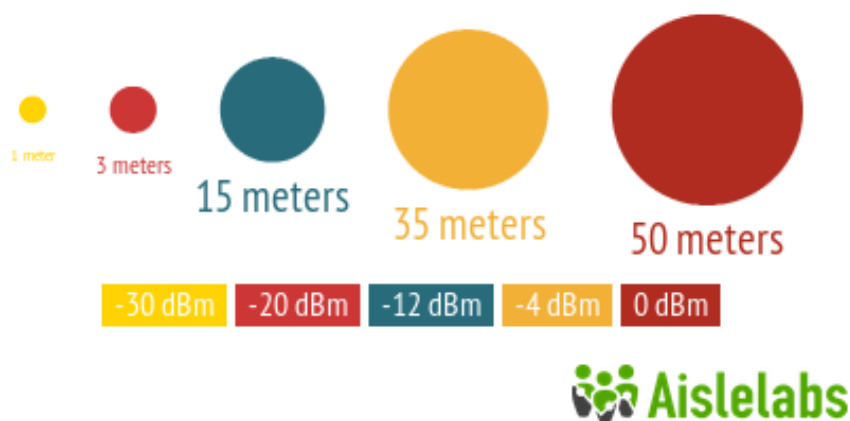
Konfiguracija iBeaconov poteka tako, da v mobilni aplikaciji proizvajalca iBeacona določimo nastavljive parametre, nato se mobilna naprava preko aplikacije poveže z iBeaconom in mu posodobi spremenjene parametre. Obstaja tudi način konfiguracije več iBeacon naprav hkrati, katerega bom predstavil v razdelku 3.4.1.

Nastavitev moči oddajanja iBeacona je najbolj odvisna od tega, kakšen domet želimo, da ima signal iBeacona. Moč signala se imenuje tx moč (angl. *tx power*), ki se meri v decibel-milivatih –  $dBm$ <sup>10</sup>. Na sliki 2.4 lahko vidimo ocenjen domet iBeacona glede na nastavitev moči oddajanja. Najvišja možna nastavitev moči oddajanja iBeacona na sliki manjka in je  $4\ dBm$ , pri kateri je ocenjen doseg signala iBeacona 70 metrov. Domet je pri nastavitvi moči na  $0\ dBm$  približno 50 metrov, pri  $-4\ dBm$  približno 35 metrov, pri  $-12\ dBm$  približno 15 metrov, pri  $-20\ dBm$  približno 3 metre in pri  $-30\ dBm$  približno 1 meter. Iz tega je razvidno, da so smiselne nastavitve moči oddajanja od  $-12\ dBm$  naprej, saj je v nasprotnem primeru domet premajhen<sup>11</sup>.

Z večanjem oglaševalnega intervala lahko podaljšamo življenjsko baterijo iBeacona, vendar lahko že malo previsok interval povzroči nezanesljivo delovanje spremljanja, ter zlasti odkrivanja. Apple predlaga interval  $100\ ms$ , kar je zelo

<sup>10</sup>Razmerje moči signala v decibelih v odnosu do enega milivata.

<sup>11</sup>Seveda obstajajo primeri, kjer bi zadoščal domet manjši kot 5 metrov, vendar je vprašanje ali je uporaba iBeacon tehnologije v tem primeru najboljša rešitev.



Slika 2.4: Domet iBeacona glede na moč oddajanja [2]

varna, vendar energetske relativno potratna nastavitve. Iz lastnih izkušenj se je najbolje držati Applevega priporočila, saj se že pri nastavitvi intervala na 500 *ms* pojavijo motnje, ki so opazne že pri spremljanju, pri odkrivanju, ki je pri določanju razdalje do iBeacona že tako precej nezanesljiv, pa še bolj. Ob uporabi spremljanja, v kombinaciji s prevelikim oglaševalnim intervalom, sem naletel na motnjo: mobilna naprava prejema dogodke o vstopu/izstopu iz regije kljub temu, da je naprava krepko znotraj dometa signala iBeacona.

Podjetje Aislelabs je opravilo obsežno raziskavo [1] o vplivu nastavitve moči oddajanja ter oglaševalnega intervala na življenjsko dobo. Test so opravili na več kot 20 iBeacon napravah različnih proizvajalcev. Na sliki 2.5 vidimo izmerjeno življenjsko dobo baterije pri iBeaconih proizvajalca Estimote, na sliki 2.6 pa proizvajalca Kontakt.io. V obeh testih je bila uporabljena baterija s kapaciteto 1000 miliamper-ur. Življenjska doba baterije je izražena v mesecih. Iz teh podatkov je razvidno, da povečanje intervala prinese skoraj linearno podaljšanje življenjske dobe baterije<sup>12</sup>. To je občutno več kot podaljšanje življenjske dobe baterije ob

<sup>12</sup>Podoben vzorec je opazen tudi pri iBeaconih drugih proizvajalcev vključenih v raziskavo.

nižanju moči oddajanja. Kljub temu je, kot že omenjeno, potrebna previdnost pri povečevanju oglaševalnega intervala, saj je lahko v nasprotnem primeru obnašanje spremljanja in odkrivanja preveč nezanesljivo za potrebe aplikacije ali sistema.

	100ms	645ms	900ms
0dbmw	4.0	14.6	27.2
-12dbmw	5.0	21.4	32.3
-20dbmw	5.6	22.8	35.6

Slika 2.5: Življenjska doba baterije Estimote iBeaconov v mesecih

	100ms	645ms	900ms
0dbmw	4.5	25.2	30.0
-12dbmw	5.4	24.3	32.1
-20dbmw	5.9	33.1	35.7

Slika 2.6: Življenjska doba baterije Kontakt.io iBeaconov v mesecih



## Poglavje 3

# iBeaconi pri izdelavi mobilnih aplikacij

### 3.1 Internet stvari (IoT)

Internet stvari (angl. Internet of Things) je pojem, ki označuje omrežje fizičnih objektov oziroma stvari, z vgrajeno elektroniko, programsko opremo, senzorji in podobnim, ki lahko izmenjujejo podatke z drugimi napravami, objekti in okoljem. IoT predstavlja razširitev interneta na fizični svet in povezovanje fizičnih in digitalnih entitet [21]. IoT z razvojem brezžičnih omrežij senzorjev, mobilnih naprav, brezžične komunikacije, mrežnih in oblačnih tehnologij hitro pridobiva na veljavnosti. Po nekaterih napovedih naj bi do leta 2020 globalno obstajalo 50 milijard naprav oziroma stvari s povezavo z internetom [18]. Tudi v primeru, da je ta številka malce pretirana, ni mogoče zanemariti vzorca, ki kaže, da bo v prihodnosti takšnih naprav vedno več. IoT omogoča razvoj pametnih storitev oziroma aplikacij, ki uporabljajo fizične objekte ali naprave za omogočanje interakcije s fizičnim svetom.

iBeacon tehnologijo lahko, kljub temu, da se s tem ne strinjajo vsi, uvrstimo med

IoT. Problem je v tem, da iBeaconi niso povezani z internetom<sup>1</sup>. Vseeno omogočajo enosmerno komunikacijo med sabo in napravo preko BLE in na ta način dopolnjujejo IoT omrežje. iBeaconi imajo velik potencial pri gradnji in razvoju IoT, saj predstavljajo način interakcije med pametno mobilno napravo in okoljem oziroma fizičnimi objekti.

## 3.2 Pomembni primeri uporabe iBeaconov

Glavna prednost pri razvoju mobilnih aplikacij, ki uporabljajo iBeacon tehnologijo, je spreminjanje pogleda ali stanja aplikacije glede na bližino (predstavljeno v razdelku 2.4). Primeri uporabe iBeaconov se zelo razlikujejo, in sicer od zelo preprostih, do zelo zapletenih. V splošnem lahko primere uporabe razdelimo v tri skupine.

1. Prva skupina je uporaba iBeaconov le za spremljanje, kdaj uporabnik oziroma njegova mobilna naprava vstopi ali izstopi iz iBeacon regije, ki predstavlja neko fizično entiteto. Takšni primeri so precej preprosti za implementacijo, saj lahko naprava z izvajanjem spremljanja regije prejema dogodke o vstopu oziroma izstopu iz regije, ter na podlagi prejetega dogodka sproži ustrezno akcijo. Konkreten primer uporabe je, da so v trgovini nameščeni iBeaconi, ki tvorijo iBeacon regijo (ali več regij, odvisno od potreb). Uporabnik si lahko namesti na mobilno napravo aplikacijo, ki jo ponuja trgovina oziroma veriga trgovin, ki uporabniku ob vstopu v regijo pošlje obvestilo o znižanjih, aktualnih ponudbah, novih kolekcijah ... Seveda takšen primer uporabe ni omejen le na trgovino, temveč lahko gre za poljubno komercialno ali nekomercialno organizacijo, ki ponuja svoje storitve. Prednosti takšnih primerov uporabe so v tem, da so enostavni za

---

<sup>1</sup>V tem primeru so mišljene naprave, ki so namenjene uporabi le kot iBeacon, oziroma beacon v splošnem. Nekateri proizvajalci namreč ponujajo tudi beacons, ki so namenjene nadziranju ostalih beaconov, ter se lahko povežejo z internetom preko WiFi. Poleg tega se lahko tudi nekatere mobilne naprave, ki se prav tako lahko povežejo z internetom, obnašajo kot beacons.



implementacijo in uporabo, saj mora imeti uporabnik nameščeno le aplikacijo. Stroški nakupa iBeaconov niso veliki, saj jih za opisano delovanje ni potrebno veliko.

2. Naslednja skupina primerov uporabe iBeacon tehnologije je spreminjanje pogleda ali stanja aplikacije glede na bližino do enega ali več iBeaconov. Takšni primeri uporabe so kompleksnejši, saj je določanje razdalje med iBeaconom in napravo težje in manj zanesljivo, kot določanje vstopa ali izstopa iz regije. Bližino lahko opredelimo z dejansko ocenjeno razdaljo v metrih ali s pomočjo bližinskih stanj (predstavljenih v 2.5.3). Takšne primere uporabe rešujemo z izvajanjem odkrivanja, ki aplikaciji daje informacije o ocenjeni razdalji oziroma bližinskem stanju iBeacona. Primer uporabe iz te skupine je, da se uporabniku prikažejo informacije o nekem objektu, ko se mu približa. V tem primeru je en objekt definiran z enim iBeaconom. Pri tem primeru je treba biti pazljiv pri poziciji iBeaconov, saj lahko v primeru, da so si med sabo preblizu, pride do konfliktov pri ugotavljanju kateri je najbližje. To pomeni, da če imamo veliko takšnih objektov, potrebujemo tudi veliko iBeaconov, kar lahko predstavlja velik strošek.
3. Zadnja skupina so primeri uporabe, kjer želimo določati mikrolokacijo uporabnika v notranjosti v realnem času, kar lahko imenujemo notranje lociranje<sup>2</sup> (angl. indoor location). Omenjeni koncept je trenutno največji izziv za razvijalce, ki uporabljajo iBeacon tehnologijo. Sistem notranjega lociranja mora biti dovolj natančen in dovolj odziven, kar predstavlja velik problem, saj je ocenjevanje razdalje pri iBeacon tehnologiji lahko hkrati nenatančno in neodzivno. V vsakem primeru je za razvoj sistema notranjega lociranja potrebno veliko število iBeaconov ter razvoj dobrega algoritma za določanje lokacije v razmerju do znanih lokacij iBeaconov. To pomeni, da je za razvoj sistema ali aplikacije za notranje lociranje potrebno veliko virov (čas, denar, razvijalci).

---

<sup>2</sup>Za koncept določanje mikrolokacije v realnem času obstaja veliko izrazov: notranja navigacija, notranja pozicija itd. Sam se bom zaradi jasnosti držal izraza notranje lociranje.

Omenjene skupine primerov uporab iBeaconov so relevantne za preproste primere uporabe. Večina realnih primerov uporabe je kombinacija preprostejših primerov uporabe iz teh skupin.

### 3.3 Primeri in analiza obstoječih aplikacij z iBeacon

Od leta 2013, ko je Apple predstavil iBeacon tehnologijo, je nastalo kar precej aplikacij, ki z uporabo iBeaconov nadgrajujejo svoje storitve. Kot ilustracijo uspešne integracije iBeacon tehnologije na trg mobilnih aplikacij sem izbral dve zanimivi aplikaciji, ki jih bom podrobneje predstavil. Aplikaciji sem izbral tako, da rešujeta različne probleme, ter zagotavljata različne funkcionalnosti.

#### 3.3.1 Launch here

Launch here [17] je aplikacija za iPhone, ki odpira aplikacije ali URL-je glede na uporabnikovo lokacijo oziroma bolj rečeno, uporabnikovo bližino do nekega objekta v prostoru. Uporabnik doma<sup>3</sup> postavi iBeacone ob različne predmete v stanovanju, nato v aplikaciji definira katera aplikacija ali URL naj se odpre, ko se iPhone približa iBeaconu ob predmetu. Torej lahko uporabnik definira pare predmetov in aplikacij. Primeri takšnih parov v uporabi bi lahko bili: ko se približam fotelju, naj se odpre aplikacija za nadziranje televizije, ko se približam postelji, naj se odpre budilka, ko se približam aparatu za kavo, naj se odpre URL z novicami itd. Uporabnik lahko sam nastavi vrednost med 0.5 in 5 metrov, ki določa kako blizu mora biti predmetu (oziroma pripadajočemu iBeaconu), da se dogodek sproži.

Aplikacija v grobem deluje tako, da uporablja odkrivanje za odkritje iBeaconov ter določanje razdalje med napravo in iBeaconom. Ko je razdalja med iBeaconom in napravo manjša kot izbrana vrednost, uporabnik prejme obvestilo, s katerim

---

<sup>3</sup>Primarno je namenjena domači uporabi, vendar lahko uporabnik sam postavi iBeacone kamor želi.

potrdi odpiranje aplikacije ali URL-ja.

Launch here je primer enostavne aplikacije, ki je intuitivna in preprosta za uporabo. Koncept odpiranja aplikacij glede na bližino predmetom je sicer zanimiv, vendar težko upraviči nakup iBeaconov, ki so potrebni za delovanje. Kljub temu je to dober primer implementacije aplikacije z uporabo iBeaconov.

### 3.3.2 Aplikacija za pomoč slabovidnim uporabnikom pri navigaciji po letališču

Podjetje Indoo.rs, ki se ukvarja z lokacijo v notranjosti, je v sodelovanju z Mednarodnim letališčem San Francisco (angl. San Francisco International Airport) razvilo sistem in mobilno aplikacijo za sistema iOS in Android, ki pomaga slabovidnim potnikom z navigacijo po letališču [14]. Gre za aplikacijo, ki med drugim nudi lokacijo oziroma navigacijo v notranjosti. Za pridobivanje podatkov o uporabnikovi lokaciji na letališču, aplikacija uporablja iBeacone in WiFi dostopne točke. Nato s kompleksnim algoritmom te podatke obdelava in skuša določiti čim bolj natančno lokacijo uporabnika. Na podlagi njegove lokacije, ga lahko aplikacija usmerja do ključnih točk na letališču. Ker je aplikacija primarno namenjena slabovidnim, omogoča tudi zvočno usmerjanje z uporabo dostopnosti<sup>4</sup> (angl. accessibility). Trenutno je razvit prototip aplikacije, na katerem poteka testiranje, torej aplikacija še ni prosto dostopna uporabnikom<sup>5</sup>.

Delovanje aplikacije je težko analizirati, saj gre za implementacijo precej kompleksnega sistema. Kljub temu je jasno, da aplikacija uporablja odkrivanje za pridobivanje podatkov o razdalji med uporabnikom in iBeaconi, in da je velik del določanja lokacije uporabnika analiza razdalj do iBeaconov, katerih lokacije so znane in umeščene v nekakšen virtualni prostor.

Razvoj takšne aplikacije zahteva veliko virov (kot omenjeno v razdelku 3.2 za primer notranjega lociranja), vendar uporabniku ni treba storiti nič drugega kot

<sup>4</sup>Funkcionalnost pametnih telefonov za pomoč ljudem s posebnimi potrebami oziroma v splošnem načrtovanje izdelkov, naprav, storitev ali okolij, ki so prijazne do ljudi s posebnimi potrebami.

<sup>5</sup>V času pisanja diplomske naloge: avgust 2015.

si na mobilno napravo prenesti mobilno aplikacijo.

### 3.3.3 Primerjava med aplikacijama

Prva aplikacija je relativno nezahtevna za implementacijo, medtem ko je druga zelo zahtevna. Po drugi strani mora uporabnik pri prvi sam priskrbeti, razporediti in vzdrževati iBeacone, kar je velik napor za tako preprosto in omejeno funkcionalnost. Aplikaciji sta po svoji zasnovi različni v smislu, da je ena enostavna za razvijalce, vendar kompleksna za uporabnika, druga pa kompleksna za razvijalce a enostavna za uporabnika. V splošnem želimo, da je aplikacija enostavna za uporabo in da se uporabniku ni treba ukvarjati z nakupom in postavitvijo iBeaconov. Ob tem se moramo vseeno zavedati, da če želimo razviti iBeacon aplikacijo za domačo uporabo, kar Launch here je, je nakup, postavitve in vzdrževanje iBeaconov z uporabnikove strani obvezno. Skrb za iBeacone s strani uporabnika pa ne prinaša le slabih elementov, saj ima uporabnik v tem primeru vso svobodo pri postavitvi svojega iBeacon omrežja. Primerjavo ključnih značilnosti lahko vidimo v tabeli 3.1.

Launch here	Navigacija po letališču
enostaven razvoj	kompleksen razvoj
malo virov za razvoj	veliko virov za razvoj
iBeaconi skrb uporabnika	iBeaconi skrb ponudnika
bližina do objekta	notranje lociranje
malo iBeaconov	veliko iBeaconov
poljubno okolje	mednarodno letališče San Francisco

Tabela 3.1: Primerjava med Launch here in aplikacijo za navigacijo po letališču

## 3.4 Postavitev velikega števila iBeaconov

Če za potrebe aplikacije, ki uporablja iBeacone, potrebujemo relativno majhno število iBeaconov, sam nakup, postavitev in vzdrževanje iBeaconov ne pomenijo prevelikega problema. Če razvijamo aplikacijo, ki omogoča iBeacon funkcionalnosti na večjem območju oziroma v zgradbi (bolnišnica, nakupovalni center, skladišče ...) in potrebujemo veliko iBeaconov, lahko nakup, postavitev in vzdrževanje iBeaconov pomenijo precejšen problem.

Točno število iBeacon naprav, ki jih potrebujemo, se seveda razlikuje od primera do primera, vendar lahko v splošnem govorimo o postavitvi velikega števila iBeaconov (angl. large scale iBeacon deployment), ko presežemo 50 iBeaconov. Večje število iBeacon naprav pomeni večji strošek in težje vzdrževanje.

### 3.4.1 Postavitev in vzdrževanje

Prvi korak pri postavitvi velikega števila iBeacon naprav je **nakup iBeaconov**. Konkretna cena iBeacona je odvisna od proizvajalca, vendar se cena osnovnih iBeaconov večine proizvajalcev giblje med 30 in 40 evri [23]. Strošek nakupa večjega števila naprav torej ni zanemarljiv, zlasti če govorimo o več sto ali več tisoč napravah. Ker pa je primerov sistemov, ki uporabljajo veliko število iBeaconov tudi v svetovnem merilu še relativno malo, je veliko proizvajalcev zelo fleksibilnih pri ceni svojih iBeacon naprav ob velikem nakupu, saj je uporaba naprav njihove znamke dobra promocija za proizvajalca. Na ta način lahko do neke mere omejimo začetne stroške nakupa iBeaconov.

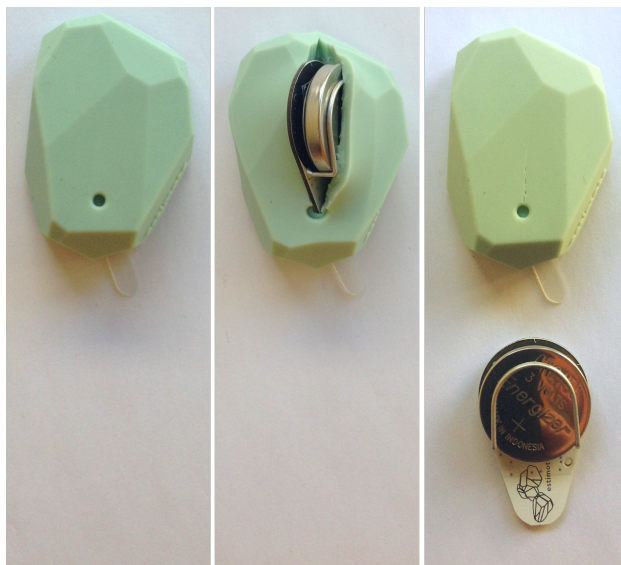
Naslednji korak je **fizična postavitev iBeaconov** po prostoru oziroma zgradbi. Lokacija posameznega iBeacona in gostota postavitve iBeaconov je odvisna od potreb aplikacije, ki jo razvijamo. Če predstavljajo iBeaconi velike regije ter želimo spremljati le vstop ali izstop iz regije, bo gostota postavitve iBeacon naprav zelo redka. Če želimo, da se aplikacija odzove glede na bližino do nekega objekta ali nekega manjšega območja, bodo iBeaconi postavljeni bolj gosto. V tem primeru moramo paziti, da niso postavljeni pregosto, saj lahko ob določanju

najbližjega iBeacona prepogosto pride do konfliktov. Če želimo določati lokacijo v notranjosti, potrebujemo zelo gosto postavitev. Za učinkovito določanje lokacije v notranjosti moramo, po priporočilu enega izmed proizvajalcev iBeaconov – Kontakt.io [15], postaviti en iBeacon na vsake 4 metre po dolžini in širini, kar je precej gosto. Jasno je, da večja gostota postavitve naprav pomeni večje število naprav.

Zadnji korak, ki je tudi najtežji, je **vzdrževanje iBeaconov**. S časom lahko iBeaconi prenehajo delovati ali jim zmanjka baterije. Ob prenehanju delovanja je potrebno iBeacon zamenjati, ob iztrošenju baterije je potrebno baterijo zamenjati. Že sama menjava baterije je lahko problem, saj je na iBeaconih nekaterih proizvajalcev postopek menjave baterije ali nemogoč ali zelo otežen. Kot primer oteženega menjavanja baterije si lahko ogledamo iBeacone proizvajalca Estimote, pri katerih moramo ob menjavi baterije prerezati zunanje gumijasto ohišje, kot lahko vidimo na sliki 3.1, s čimer pustimo vidno zarezo na ohišju [28]. Menjavanje baterije je veliko cenejše, kot menjavanje cele naprave, zato se je pri nakupu iBeaconov nekega proizvajalca dobro pozanimati, ali njihove naprave podpirajo menjavanje baterije.

Za učinkovito vzdrževanje želimo vedeti vnaprej, kdaj bo iBeaconu zmanjkalo baterije in čimprej izvedeti, da je iBeacon prenehal delovati. Tukaj naletimo na težavo, saj iBeacon tehnologija ne ponuja načina opazovanja stanja baterije in stanja delovanja iBeacona. Rešitev tega problema zato ponuja proizvajalec Kontakt.io s svojim oblračnim beaconom (angl. Cloud Beacon). Slednji je namenjen konfiguraciji in nadziranju večje skupine iBeacon naprav v doletu 200 metrov. Preko oblračnega beacona, ki je tudi povezan z internetom preko WiFi vmesnika, lahko spremljamo stanje baterije in stanje delovanja iBeaconov v njegovem dosegu. Prav tako lahko hkrati spremenimo parametre (UUID, major in minor števili, moč oddajanja in interval oddajanja) iBeaconov, nad katerimi ima nadzor, namesto da spreminjamo parametre posameznega iBeacona. Na ta način lahko iBeaconom v času, ko se ne uporabljajo, nastavimo manjšo moč oddajanja in večji interval oddajanja, zato da podaljšamo življenjsko dobo baterije.

Pomanjkanje podpore za upravljanje z veliko iBeaconi je pomanjkljivost iBeacon tehnologije, kar predstavlja velik problem pri vzdrževanju velikega števila iBeaconov. Zanašanje na proizvajalce iBeaconov, da bodo poskrbeli za funkcionalnost upravljanja z več iBeaconi je nezanesljivo, zato bi moral Apple v prihodnosti zagotoviti podporo omenjeni funkcionalnosti, če želi, da se iBeacon tehnologija uporablja tudi pri večjih projektih.



Slika 3.1: Menjava baterije pri Estimote iBeaconu [28]

### 3.4.2 Uporaba v zdravstvu

Ko govorimo o primerih uporabe iBeacon tehnologije, ne moremo mimo primerov uporabe v zdravstvu. Zaradi informatizacije in modernizacije zdravstva lahko integriramo iBeacon tehnologijo v zdravstvene informacijske sisteme v obliki mobilnih aplikacij. Motivacija za implementacijo rešitev, ki uporabljajo iBeacon tehnologijo, je poenostavitev in izboljšanje zdravstvenih storitev. To lahko dosežemo z načrtovanjem aplikacij, ki so namenjene uporabnikom zdravstvenih storitev ali osebam, zaposlenim v zdravstvu (zdravniki, delavci v zdravstveni negi, reševalci ...). Primer aplikacije, ki je namenjena uporabnikom zdravstvenih storitev, je aplikacija, ki ob prihodu v ambulantno uporabniku sporoči, kako dolga je čakalna

vrsta. Primer aplikacije, ki je namenjena zdravstvenim delavcem, je aplikacija, ki zdravniku, ob njegovem prihodu v ambulantno, izpiše vse naročene paciente.

Pri razvoju zdravstvenih aplikacij za uporabo v bolnišnicah ali zdravstvenih domovih gre v večini primerov za postavitev velikega števila iBeaconov, saj so zdravstvene ustanove praviloma velike stavbe, torej je treba z iBeaconi pokriti veliko prostora.

Zdravstvena panoga ponuja veliko izzivov za iBeacon tehnologijo, izmed katerih sem dva že omenil. Še nekaj zanimivih primerov potencialnih aplikacij, ki uporabljajo iBeacone:

- sledenje opreme – aplikacija, ki prikaže lokacijo nekega kosa opreme, na podlagi bližine kosa opreme do neke ključne točke. Takšna aplikacija lahko pripomore k manjšemu izgubljanju bolnišnične opreme in lažjemu iskanju opreme. Primeri bolnišnične opreme so vozički in bolniške postelje;
- navigacija po bolnišnici – aplikacija, ki lahko določa uporabnikovo lokacijo v notranjosti in mu daje napotke za pot do izbranega oddelka, ambulante ali poljubne entitete. Določanje lokacije je kompleksen problem in potrebuje veliko virov za razvoj (kot opisano v razdelkih 3.2 in 3.4.1), zato zdravstvena ustanova potrebuje veliko sredstev za takšno rešitev;
- prikaz podatkov pacienta – aplikacija, ki prikaže podatke o pacientu, ko se mu zdravstveni delavec približa ali ko vstopi v sobo. Takšna aplikacija pohitri dostop do podatkov zdravnika oziroma zdravstvenega delavca in izboljša njihovo preglednost.

Aplikaciji, ki sem ju razvil in sta predstavljeni v poglavju 4, temeljita na zadnjem primeru – prikaz podatkov pacienta.



## Poglavje 4

# Razvoj aplikacije za prikaz podatkov pacienta na platformah Android in iOS

### 4.1 Opis in cilj razvitih aplikacij

#### 4.1.1 Opis

Razviti aplikaciji omogočata prikaz podatkov ležečega pacienta na podlagi bližine do pacienta. To v praksi pomeni, da ko se zdravstveni delavec z mobilno napravo približa pacientu, se mu na njej prikažejo podatki tega pacienta. Ta funkcionalnost je dosežena z uporabo iBeacon tehnologije. Ležeči pacient oziroma bolje rečeno postelja, na kateri leži pacient, je predstavljena z enim iBeaconom. Aplikacija spremlja iBeacone okoli sebe in prikaže podatke najbližjega pacienta, predstavljenega z najbližjim iBeaconom.

#### 4.1.2 Cilji

Temeljni cilj razvoja aplikacij ni razvoj aplikacij, ki bi se lahko uporabili v bolnišnici v praksi, saj sta takšni aplikaciji zelo kompleksni in je njun razvoj dolgotrajen ter zahteva večjo ekipo. Temeljni cilj razvoja aplikacij je namreč raziskava, kako

lahko iBeacon tehnologijo vključimo v razvoj mobilnih aplikacij oziroma bolj specifično razvoj mobilnih zdravstvenih aplikacij. Razviti aplikaciji sta torej dokaz koncepta (angl. proof of concept) uporabe bližine za spreminjanje delovanja aplikacije, z uporabo iBeaconov. V tem primeru to pomeni prikazovanje podatkov pacienta.

Naslednji cilj razvoja aplikacij je primerjava razvoja aplikacije, ki uporablja iBeacone glede na mobilno platformo, in sicer glede na glavni mobilni platformi – Android in iOS. Glede na to, da iBeacon tehnologija v osnovi nudi polno podporo le iOS platformi, je zlasti zanimivo, kako omejen je razvoj takšnih aplikacij na Android platformi. Zato sem se odločil za razvoj aplikacije na obeh platformah. Zadnji cilj razvoja aplikacij je preizkus iBeacon tehnologije in iBeacon naprav v praksi. Na ta način lahko podam svoje mnenje o uporabnosti in zanesljivosti iBeacon tehnologije, glede na lastne izkušnje.

## 4.2 Uporabljene tehnologije, razvojna orodja in knjižnice

### 4.2.1 Android aplikacija

#### Java

Java [13] je objektno usmerjen programski jezik, ki ga je za podjetje Sun Microsystems razvil James Gosling. Verzija 1.0 je izšla leta 1996, aktualna verzija pa je 8.0. Sintaksa jave izvira iz programskih jezikov C in C++, vendar omogoča manj nizkonivojskih funkcionalnosti kot omenjena programska jezika. Glavni različici jave sta:

- Java Standard Edition – Java SE, ki se uporablja za razvoj računalniških aplikacij in javanskih aplikacij;
- Java Enterprise Edition – Java EE, ki se uporablja za razvoj kompleksnejših aplikacij, za uporabo v večjih podjetjih.

## 4.2. UPORABLJENE TEHNOLOGIJE, RAZVOJNA ORODJA IN KNJIŽNICE

Java je eden najpopularnejših programskih jezikov in osrednji programski jezik za razvoj Android mobilnih aplikacij.

### **XML – Extensible Markup Language**

XML [8] je razširljiv označevalni jezik (angl. markup language), uporabljen za kodiranje dokumentov in je zasnovan tako, da je berljiv za ljudi in stroje. XML je bil zasnovan za uporabo na internetu in ponavadi predstavlja podatkovne strukture uporabljene pri spletnih storitvah. XML se pri razvoju Android aplikacij uporablja za deklaracijo in oblikovanje elementov grafičnega vmesnika.

### **Android Studio**

Android Studio [31] je integrirano razvojno okolje (angl. integrated development environment – IDE), ki je namenjen razvoju Android mobilnih aplikacij. Razvil ga je podjetje Google, vendar je zasnovan na osnovi IntelliJ IDEA, ki je IDE za razvoj javanskih aplikacij razvit s strani podjetja JetBrains. Android Studio uporablja Gradle kot sistem za gradnjo programja (angl. build system). Uporabil sem ga za razvoj Android aplikacije.

### **Android SDK**

Android SDK [30] je SDK, ki vsebuje obsežno skupino razvojnih orodij, ki med drugim vključujejo razhroščevalnik (angl. debugger), knjižnice, simulator Android mobilne naprave in dokumentacijo. Uradno podprta IDE-ja sta Eclipse in Android Studio.

### **Android Beacon knjižnica**

Android Beacon knjižnica [3] je knjižnica, ki omogoča uporabo beaconov na Android napravah. Razvil jo je podjetje Radius Networks, ki je odgovorno za razvoj AltBeacon standarda. Knjižnica v osnovi zaznava beacone, ki ustrezajo AltBeacon standardu, vendar jo lahko prilagodimo, da zaznava tudi druge beacone (tudi

iBeacone). Knjižnico sem uporabil za zaznavanje iBeaconov, pri razvoju Android aplikacije.

### **Spring ogrodje**

Spring ogrodje [26] (angl. framework) je aplikacijsko ogrodje za Java platformo. Jedrne funkcije ogrodja lahko uporablja katerakoli javanska aplikacija. Knjižnico sem uporabil kot orodje za pošiljanje HTTP zahtevkov.

## **4.2.2 iOS aplikacija**

### **Swift**

Swift [5] je programski jezik, ki ga je razvil Apple in se uporablja za razvoj iOS, OS X in WatchOS<sup>1</sup> aplikacij. Za swift je značilnih več paradigem programskih jezikov: protokolno programiranje, objektno programiranje, funkcijsko programiranje, imperativno programiranje in bločno programiranje. Prva verzija swifta je izšla leta 2014, najnovejša verzija – 2.0 bo izšla jeseni, leta 2015 (trenutno, septembra 2015 je še v procesu beta testiranja). Podjetje Apple je naznanilo, da bo Swift 2 odprtokoden in bo podprt na iOS, OS X in Linux platformah. Zasnovan je bil kot varnejša in modernejša alternativa Objective-C-ju<sup>2</sup>, z bolj pregledno sintakso.

### **XCode**

XCode [22] je integrirano razvojno okolje, krajše IDE, namenjen razvoju aplikacij za iOS, OS X in WatchOS platforme. Razvoj aplikacij za te platforme je možen le s pomočjo XCode IDE-ja. Razvil ga je Apple in deluje le na operacijskem sistemu Mac OS X. XCode je ekvivalent Android Studio za razvoj iOS mobilnih aplikacij (s tem da lahko v XCode razvijamo tudi aplikacije za OS X in WatchOS,

---

<sup>1</sup>Operacijski sistem za Appleovo pametno uro – Apple Watch.

<sup>2</sup>Objektno orientiran programski jezik. V preteklosti se je uporabljal kot glavni programski jezik za razvoj na operacijskih sistemih iOS in OS X in ga v tej vlogi počasi nadomešča Swift.

## 4.2. UPORABLJENE TEHNOLOGIJE, RAZVOJNA ORODJA IN KNJIŽNICE

Android Studio je pa namenjen izključno razvoju Android aplikacij). XCode sem uporabil za razvoj iOS aplikacije.

### **iOS Core Location ogrodje**

iOS Core Location ogrodje [4] je ogrodje, vključeno v mobilni operacijski sistem iOS, ki omogoča določanje lokacije in usmeritve iOS naprave. Lokacijo oziroma blizino v primeru iBeaconov, določa z uporabo strojne opreme (GPS, WiFi, BLE in iBeacon), ki je na voljo. Ogrodje sem uporabil za interakcijo med mobilno napravo in iBeaconi.

### **4.2.3 Obe aplikaciji**

#### **EhrScape**

EhrScape [19] je odprta platforma zdravstvenih podatkov, ki ponuja aplikacijski programski vmesnik (angl. application programming interface – API), s pomočjo katerega lahko dostopamo do ogromne baze zdravstvenih zapisov. EhrScape je namenjen olajšanju razvoja zdravstvenih aplikacij in storitev. Sam sem v obeh aplikacijah za vir podatkov o pacientih uporabil EhrScape-ov API za dostop do baze zdravstvenih zapisov.

### **4.2.4 Strežnik**

#### **Node.js**

Node.js [27] je odprtokodno izvajalno okolje za strežniške aplikacije. Nudi podporo za številne platforme, med drugim za OS X, Microsoft Windows in Linux. Node.js aplikacije so napisane v programskem jeziku JavaScript<sup>3</sup>. S pomočjo Node.js lahko zelo hitro in preprosto razvijemo osnovno strežniško aplikacijo, kot sem jo potreboval sam za svoji mobilni aplikaciji.

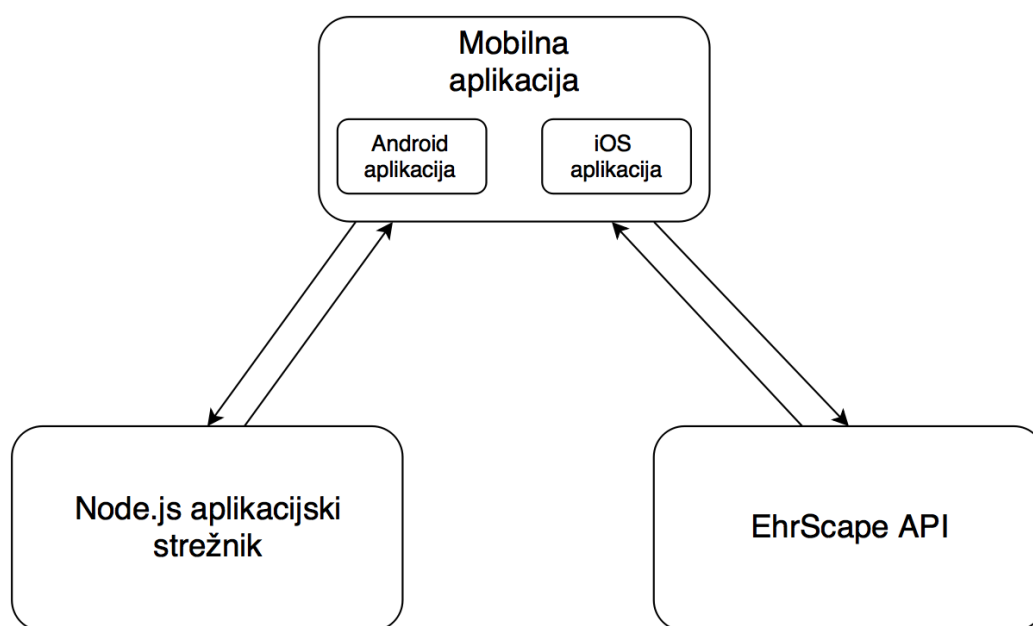
---

<sup>3</sup>Visoko-nivojski, dinamičen, netipiziran programski jezik, ki se tipično uporablja za razvoj spletnih vsebin.

## 4.3 Tehničen opis delovanja aplikacij

Razviti sta iOS in Android aplikacija. Funkcionalnost obeh aplikacij je prikazovanje podatkov najbližjega pacienta, ki je določen z uporabo iBeacon tehnologije. Motivacija za razvoj aplikacij je preizkus iBeacon tehnologije pri realnem razvoju mobilne aplikacije na platformah Android in iOS ter primerjava razvoja iBeacon aplikacije glede na platformo. Razviti aplikaciji služita kot osnova za potencialen razvoj razširjene mobilne aplikacije za uporabo v bolnišničnem okolju. Aplikaciji delujeta skoraj enako, zato bom ob opisovanju delovanja govoril le o eni aplikaciji.

### 4.3.1 Delovanje glavnih funkcionalnosti aplikacij



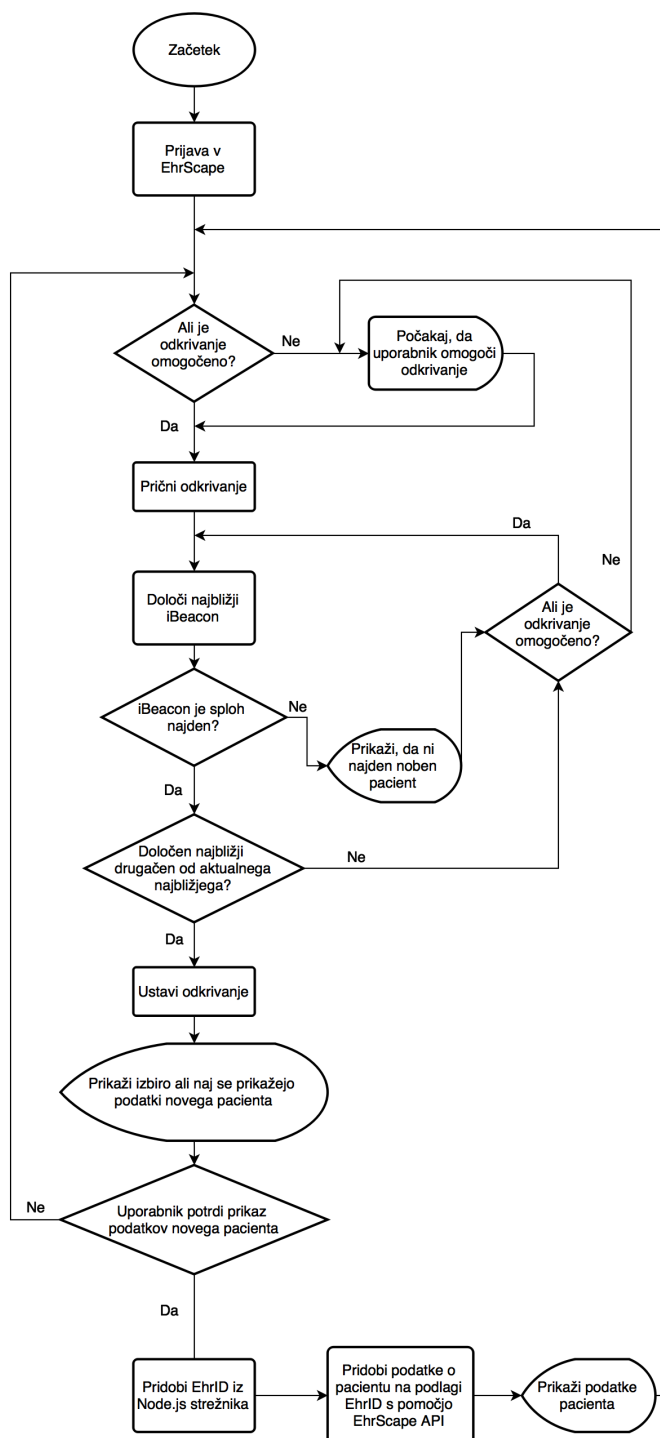
Slika 4.1: Diagram, ki predstavlja več-nivojsko arhitekturo, na kateri temelji delovanje aplikacije

Za delovanje aplikacije je potrebna več-nivojska arhitektura. Elemente arhitekture in relacije med njimi vidimo na diagramu na sliki 4.1. Elementi so naslednji:

- mobilna aplikacija, ki opazuje iBeacone v bližini in išče najbližjega oziroma skupino najbližjih in prikazuje informacije o najbližjem pacientu oziroma pacientih;
- Node.js strežnik, ki hrani pare, ki povezujejo iBeacon s pacientom. Prva komponenta para je zaporedna združitev oglaševalnih podatkov iBeacona (UUID + glavno število + drugotno število). Druga komponenta para je EhrID, ki je enoličen identifikator pacienta;
- EhrScape baza kot storitev z API-jem, ki omogoča pridobitev podatkov o pacientu na osnovi EhrID-ja.

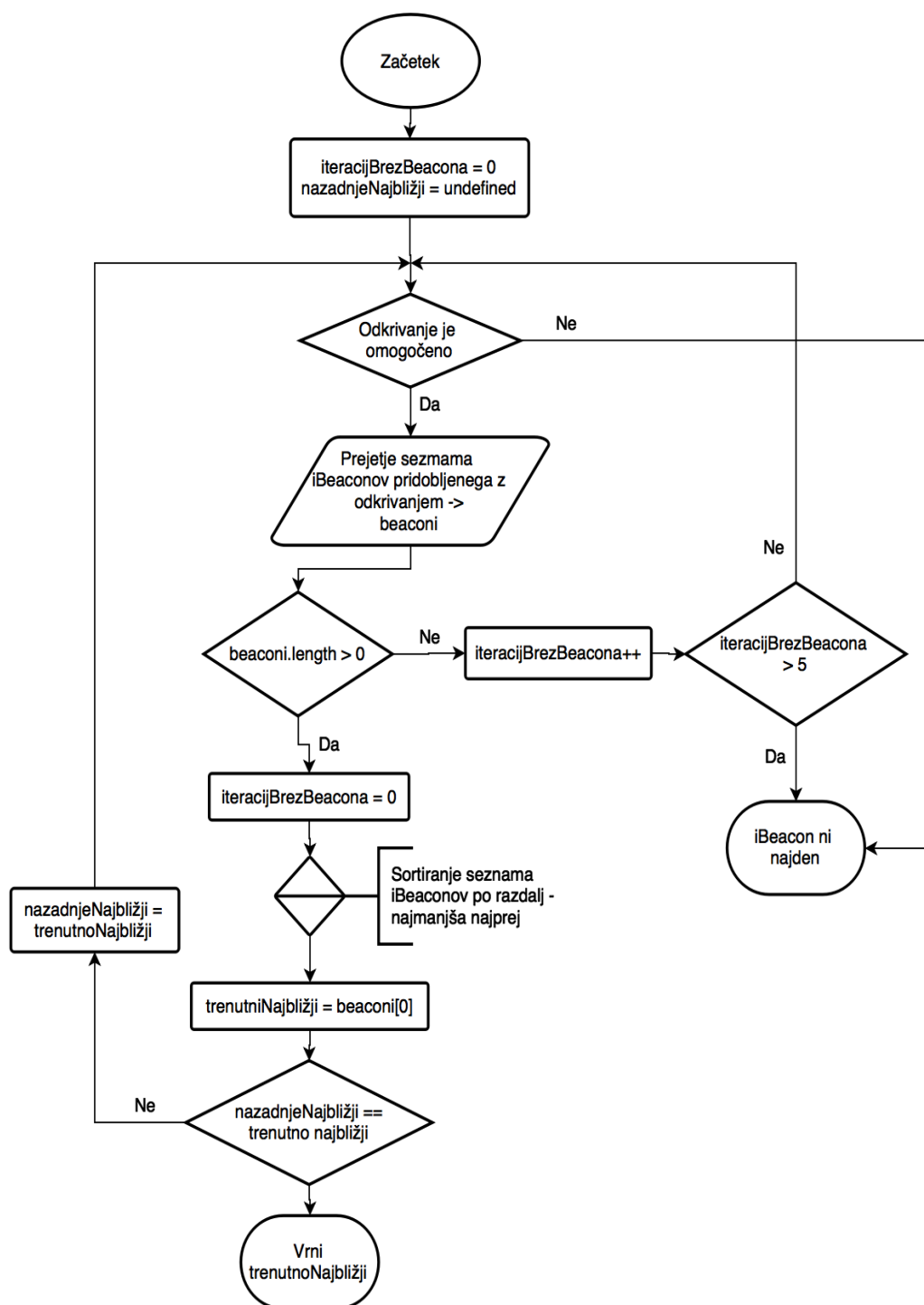
Mobilna aplikacija pošilja zahteve in prejema odgovore iz Node.js aplikacijskega strežnika in EhrScape baze. Node.js in EhrScape od aplikacije prejemata zahteve in pošiljata odgovore aplikaciji. Med njima komunikacija direktno ne poteka.

Delovanje aplikacije je sledeče. Na začetku aplikacija, z uporabo referenčnega uporabniškega imena in gesla, izvede prijavo v EhrScape. Aplikacija izvaja odkrivanje in z njim določa najbližji iBeacon. En iBeacon predstavlja enega pacienta. Sedaj mora ugotoviti, na podlagi podatkov izbranega iBeacona, za katerega pacienta gre. Zato zlepljene oglaševalne podatke v obliki HTTP POST zahtevka pošlje na Node.js strežnik, ki aplikaciji kot odgovor pošlje pripadajoč EhrID (v primeru da na Node.js strežniku obstaja vnos za poslane podatke iBeacona). Nato z EhrScape API-jem (z uporabo HTTP GET zahtevkov) pridobi nekaj podatkov o pacientu in jih prikaže na zaslonu. Če se kakšen drug iBeacon bolj približa mobilni napravi, aplikacija obvesti uporabnika, da je zaznala bližjega pacienta. Nato lahko uporabnik potrdi ogled podatkov novega najbližjega pacienta, ali pa ohrani pogled na podatkih starega pacienta. Uporabnik lahko izklopi izvajanje odkrivanja, če želi imeti nemoten fokus na trenutno prikazanem pacientu. Prikaz delovanja v obliki diagrama zaporedja lahko vidimo na sliki 4.2.



Slika 4.2: Diagram zaporedja, ki predstavlja delovanje aplikacije





Slika 4.3: Diagram zaporedja, ki predstavlja določanje najbližjega iBeacona

Na diagramu zaporedja na sliki 4.2 je določanje najbližjega iBeacona označeno le z eno operacijo. Podrobnejši prikaz določanja najbližjega iBeacona v obliki diagrama zaporedja vidimo na sliki 4.3, opis omenjenega delovanja je pa naslednji. Z izvajanjem odkrivanja na približno vsako sekundo, prejme seznam vidnih iBeacon objektov. Posamezen dogodek, ko prejme seznam vidnih iBeacon objektov, bom imenoval iteracija odkrivanja. Vsakič, ko se to zgodi, iz seznama izbere tistega, ki ima najmanjšo ocenjeno razdaljo. Zaradi nezanesljivosti ocenjene razdalje in možnosti, da kdaj kateri izmed iBeaconov v dometu ni viden v iteraciji odkrivanja, aplikacija zahteva strožji pogoj za to, da iBeacon zares prepozna kot najbližji. Ta pogoj je, da mora imeti iBeacon v dveh iteracijah odkrivanja zaporedoma najmanjšo razdaljo. Na ta način izgubimo na odzivnosti prepoznavanja najbližjega iBeacona, vendar se zavarujemo pred primeri, kjer je naprava bližje enemu iBeaconu, vendar na podlagi (nezanesljivih) podatkov presodi, da je bližje drugemu. Kljub temu včasih prihaja do konfliktov pri določanju najbližjega iBeacona. Zaradi tega je implementirana možnost izklopa odkrivanja, da lahko uporabnik ohrani pogled na trenutnem pacientu, brez prejemanja novih obvestil, da je bil odkrit bližji pacient.

V android aplikaciji uporabljamo za interakcijo z iBeaconi Android Beacon knjižnico, ki z osnovnimi nastavitvami zaznava AltBeacone. Da zaznava iBeacone, moramo spremeniti konfiguracijo z naslednjo vrstico javanske kode, kjer je *beaconManager* instanca objekta *BeaconManager*:

```
beaconManager.getBeaconParsers().add(new BeaconParser()  
    .setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,  
    i:22-23,p:24-24,d:25-25"));
```

Izvleček javanske kode, ki prikazuje kako zgleda funkcija, ki vsako sekundo prejme seznam vidnih iBeaconov pri Android aplikaciji, kjer je *beaconManager* instanca objekta *BeaconManager*:

```
@Override
public void onBeaconServiceConnect() {
    beaconManager.setRangeNotifier(new RangeNotifier() {
        @Override
        public void didRangeBeaconsInRegion(Collection<Beacon>
            beacons, final Region region) {
            // tukaj aplikacija ustrezno odreagira glede
            // na seznam iBeaconov imenovan beacons
        }
    })
}
```

Izvleček Swift kode, ki prikazuje kako zgleda funkcija, ki vsako sekundo prejme seznam vidnih iBeaconov pri iOS aplikaciji:

```
func locationManager(manager: CLLocationManager!,
    didRangeBeacons beacons: [CLBeacon]!,
    inRegion region: CLBeaconRegion!) {
    // tukaj aplikacija ustrezno odreagira glede na
    // seznam iBeaconov imenovan beacons
}
```

### 4.3.2 Delovanje grafičnega vmesnika aplikacij

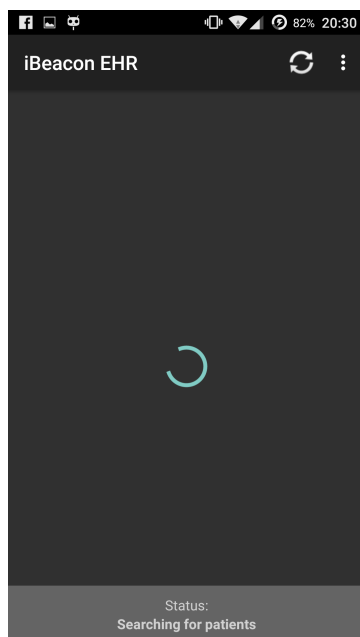
Na slikah 4.4 in 4.6 lahko vidimo prikaz zaslonskega stanja obeh aplikacij. V spodnjem delu zaslona se izpisuje trenutni status aplikacije. Možni statusi so:

- iskanje pacienta,
- pacient ni bil najden – v primeru, da v petih iteracijah odkrivanja ni viden niti en iBeacon,

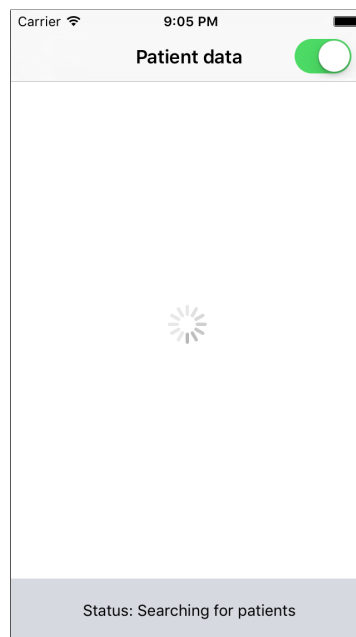
- pridobivanje EhrID-ja iz Node.js,
- pridobivanje podatkov o pacientu s pomočjo EhrScape API-ja,
- iskanje zaključeno – pacient je bil najden in prikazan.

V desnem zgornjem kotu se nahaja gumb za izklop odkrivanja. Pri iOS aplikaciji je to gumb, ki ima le dve stanji: vključeno, izključeno. Pri Android aplikaciji mora uporabnik pritisniti gumb za nastavitve (skrajni desni), nato se pokaže možnosti izklopa odkrivanja, kot je prikazano na sliki 4.5.

Medtem ko aplikacija išče najbližji iBeacon, je v sredini aplikacije viden indikator aktivnosti (angl. activity indicator), ki uporabniku sporoča, da je iskanje najbližjega pacienta v teku. To lahko vidimo na sliki 4.4. Ko aplikacija določi najbližjega pacienta in pridobi podatke o njem, se ti pokažejo na zaslonu, kot prikazano na sliki 4.6.

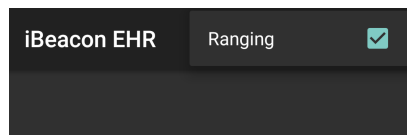


(a) Android aplikacija

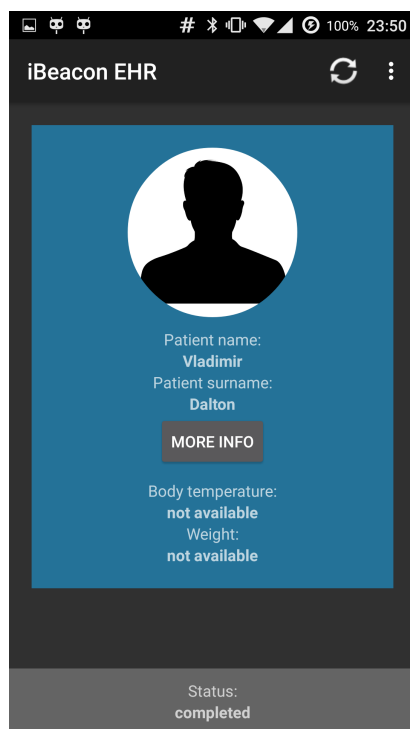


(b) iOS aplikacija

Slika 4.4: Prikaz zaslonske slike aplikacij med iskanjem najbližjega pacienta



Slika 4.5: Prikaz možnosti izklopa odkrivanja pri Android aplikaciji



(a) Android aplikacija



(b) iOS aplikacija

Slika 4.6: Prikaz zaslonske slike aplikacij ob prikazu podatkov najbližjega pacienta

## 4.4 Primerjava razvoja iBeacon aplikacij na platformah Android in iOS

Glede na to, da je iBeacon tehnologija razvita s strani podjetja Apple (produkt katerega je tudi mobilni operacijski sistem iOS) in je v osnovi podprta le na iOS platformi, je resnično vprašanje kako omejen je razvoj iBeacon aplikacij na Androidu. Na podlagi lastnih izkušenj z razvojem ekvivalentne aplikacije za obe platformi lahko izpostavim dva večja problema razvoja iBeacon aplikacij na Android platformi.

Prvi problem je že večkrat izpostavljeno dejstvo, da Android platforma v osnovi ne nudi podpore za interakcijo z iBeaconi. Odvisnost od SDK-ja ali knjižnice zunanjega ponudnika je nezanesljiva, saj lahko ponudnik preneha nuditi podporo za svoj SDK oziroma knjižnico. Slednje lahko predstavlja problem za razvijalca aplikacije, saj ne more uporabiti novih funkcij iBeacon tehnologije oziroma mora poiskati alternativo drugega zunanjega ponudnika. Pomanjkanje podpore iBeacon tehnologije na Androidu posredno povzroča, da je glavni fokus proizvajalcev iBeaconov pri razvoju programske opreme<sup>4</sup> iOS platforma.

Drug problem Android platforme je manjša natančnost pri izvajanju odkrivanja. Do tega problema pride, ker BLE pregledovanje (angl. scanning) deluje drugače na Androidu, kot na iOS-u. Na Androidu namreč deluje tako, da si naprava zapomni le eno meritev moči signala na en pregled (angl. scan), na iOS-u pa tako, da si naprava zapomni poljubno število meritev moči signala na en pregled. To pomeni, da če imamo iBeacon, ki oddaja svoje oglaševalne podatke desetkrat na sekundo in naprava izvaja pregled enkrat na sekundo, bo imela Android naprava po desetih sekundah na voljo 10 meritev, iOS naprava pa 100 meritev [29].

Kljub predstavljenima problemoma, je razvoj Android iBeacon aplikacij precej podoben razvoju iOS iBeacon aplikacij in ni nič manj smiseln.

---

<sup>4</sup>V mislih imamo programsko opremo, razvito s strani proizvajalcev iBeaconov, ki je namenjena za konfiguracijo in interakcijo z lastnimi iBeaconi.

## 4.5 SWOT analiza

Gre za metodo, ki določi prednosti, slabosti, priložnosti in grožnje (angl. strengths, weaknesses, opportunities and threats – SWOT) produkta, osebe, organizacije, industrije in še česa drugega. Elemente SWOT analize lahko predstavimo v obliki dvodimenzionalne kvadratne matrike. Lahko jih uredimo v stolpce in vrstice matrike, glede na značilnosti. V levem stolpcu naj bodo elementi, ki predstavljajo pozitivne lastnosti (prednosti in priložnosti), v desnem pa elementi, ki predstavljajo negativne lastnosti (slabosti in grožnje). V zgornji vrstici naj bodo elementi, ki so notranjega izvora (prednosti in slabosti), v spodnji pa elementi, ki so zunanjega izvora (priložnosti in grožnje). Tako dobimo pregleden prikaz, katerega shemo lahko vidimo na sliki 4.7

SWOT analiza je opravljena za aplikacijo, ki predstavlja razširitev razvitih aplikacij. To pomeni, da gre za aplikacijo, ki za jedro vzame funkcionalnost razvitih aplikacij in doda ter razširi ostale potrebne komponente za uporabo v bolnišničnem okolju. Nekaj primerov takšnih dodatkov in razširitev: implementacija pravega sistema za prijavo, izboljššan grafični vmesnik, razširitev funkcionalnosti strežnika, obsežnejši prikaz pacientovih podatkov, možnost vpisovanja pacientovih podatkov, bolj robusten algoritem za odkrivanje najbližjih iBeaconov in implementacija varnosti ob interakciji z iBeaconi.

### 4.5.1 Prednosti

Glavna prednost aplikacije za prikazovanje pacientovih podatkov na podlagi bližine je v tem, da olajša delo zdravstvenih delavcev. Aplikacija omogoča prikaz in vpisovanje pacientovih podatkov na mobilni napravi, brez iskanja pacienta. To precej skrajša sam postopek prikaza in vpisovanja podatkov v primerjavi z običajnim pristopom – podatki pacienta so prikazani na analognem pacientovem kartonu, vpisovanje pa poteka analogno v kartoteko pacienta ali digitalno preko aplikacije na računalniku. Interakcija s podatki pacienta je mogoča z uporabo le ene naprave, ki je poleg tega tudi mobilna.

Olajšanje dela zdravstvenih delavcev prinaša učinkovitejše in bolj sproščeno de-

lovno okolje. Posledice izboljšanega delovnega okolja niso omejene le na zaposlene v bolnišnici, ampak tudi na paciente oziroma uporabnike zdravstvenih storitev, saj se na ta način lahko skrajšajo čakalne dobe, poveča učinkovitost zdravljenja in izboljša interakcija med pacienti in zaposlenimi.

#### 4.5.2 Slabosti

Večina slabosti aplikacije je povezana z uporabo iBeaconov. Kljub potencialnim izboljšavam logike določanja najbližjega iBeacona, lahko še vedno prihaja do konfliktov pri prikazu najbližjega pacienta. To je posledica tega, da je ocena razdalje med iBeaconom in napravo enostavno premalo zanesljiva. Če naprava ne more določiti ustreznega najbližjega pacienta, moramo pacienta poiskati s pomočjo iskalnika pacientov (če je slednji implementiran) oziroma kakšne druge metode. Pogostost omenjenih konfliktov lahko zmanjšamo s prilagoditvijo aplikacije, da namesto enega pacienta prikaže več najbližjih pacientov, ali da iBeacone razporedimo manj gosto (kar lahko dosežemo tako, da en iBeacon predstavlja sobo in ne pacienta).

Postavitev in vzdrževanje iBeaconov predstavljata v takšnem okolju precejšen strošek in napor, saj gre za postavitev velikega števila iBeaconov (kot predstavljeno v razdelku 3.4.1). Vzdrževanje iBeaconov je težko, saj iBeacon tehnologija ne ponuja rešitve, kako hkrati spreminjati parametre ali kako učinkovito spremljati stanja baterije večjega števila iBeaconov. Problemi in rešitve vzdrževanja večjega števila iBeaconov so predstavljeni v razdelku 3.4.1.

#### 4.5.3 Priložnosti

V zdravstvu poteka proces informatizacije, kar pomeni, da se med drugim zdravstveni podatki in možnosti interakcije z njimi premikajo iz analognih oblik v digitalne. To ustvarja potrebo po razvoju kvalitetnih računalniških zdravstvenih aplikacij, med katere spadajo tudi mobilne aplikacije. Po drugi strani se trudijo bolnišnice izboljšati svoje zdravstvene storitve. Kombinacija teh dveh trendov ustvarja ugodno okolje za razvoj pametne mobilne aplikacije za prikazovanje po-



datkov o pacientu, ki uporablja iBeacon tehnologijo.

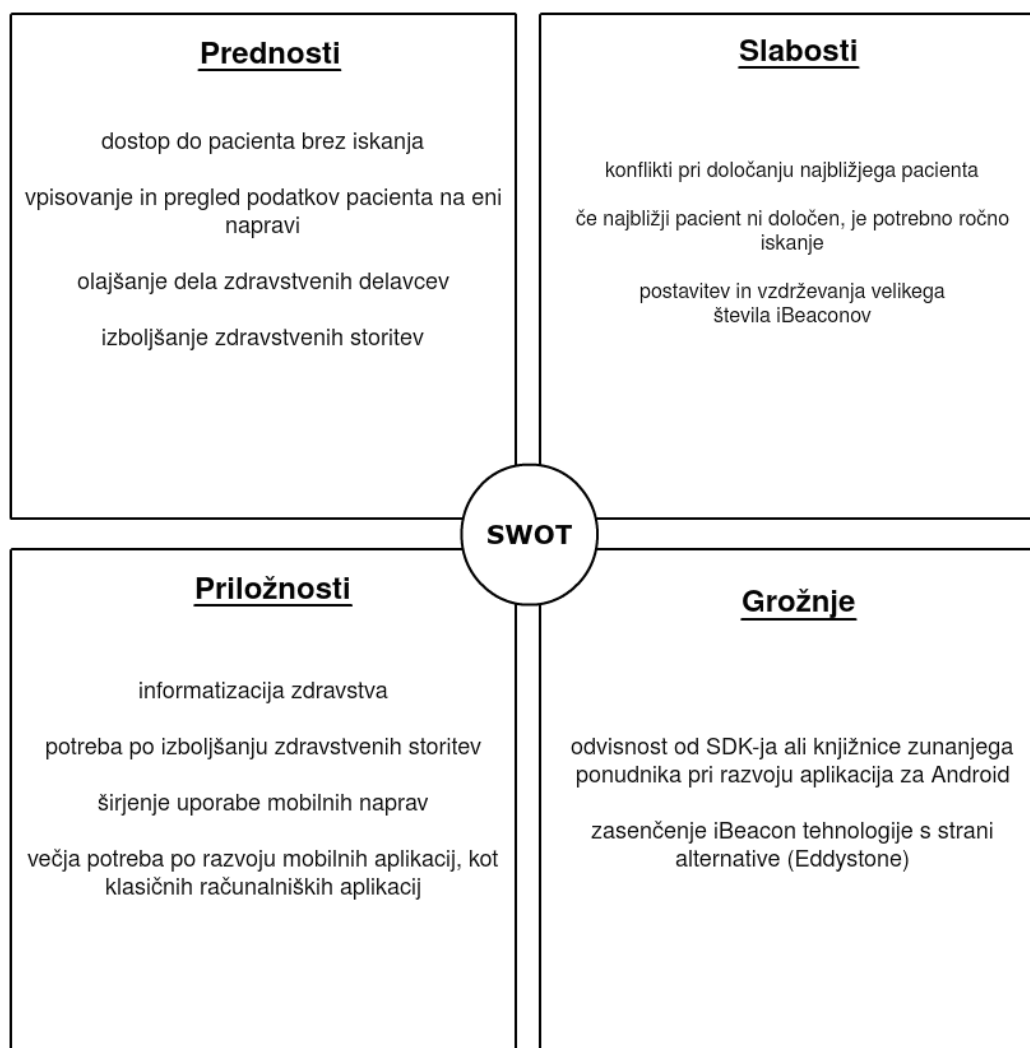
Pametne mobilne naprave (pametni telefoni, tablice, pametne ure) počasi prevzemajo vlogo najbolj pomembne in razširjene naprave za posameznika. V tej vlogi so nadomestile osebni računalnik. Večja razširjenost uporabe mobilnih naprav ustvarja potrebo po razvoju mobilnih aplikacij. Glede na trend povečevanja števila uporabnikov mobilnih naprav, ki je razviden iz podatka, da je v letu 2014 število uporabnikov mobilnih naprav preseglo število uporabnikov namiznih računalnikov [25], je potreba po razvoju mobilnih aplikacij večja, kot potreba po razvoju klasičnih računalniških aplikacij.

#### 4.5.4 Grožnje

V primeru uporabe iBeacon tehnologije pri razvoju aplikacije za prikazovanje podatkov pacienta oziroma pri razvoju kakršnekoli aplikacije, se moramo zavedati nekaterih groženj, ki so zunanjega izvora, torej nanj nimamo neposrednega vpliva. Pri razvoju Android aplikacije, ki uporablja iBeacone, Android platforma ne zagotavlja načina interakcije z iBeaconi. Problemi, ki lahko nastanejo kot posledica tega so opisani v razdelku 4.4.

Problem lahko nastane tudi v primeru, da razvijalec aplikacije kupi iBeacone nekega proizvajalca, nato pa jih ta čez nekaj časa preneha izdelovati. Razvijalcu nato preostaneta dve možnosti: menjava vseh iBeaconov ali uporaba iBeaconov različnih proizvajalcev. Nobena od možnosti ni ugodna, saj lahko obe ustvarita dodatne stroške.

Potencialno grožnjo predstavlja tudi možnost, da ena izmed alternativnih tehnologij zasenči iBeacon tehnologijo. Pri tem imamo v mislih zlasti Googlovo tehnologijo Eddystone, predstavljeno v razdelku 2.3. To bi lahko pomenilo, da bi se razvoj in podpora iBeacon tehnologije s strani Apple ustavila, kar povzroči stagnacijo razvoja novih funkcionalnosti aplikacije.



Slika 4.7: SWOT analiza aplikacije za prikazovanje podatkov pacient v matrični predstavitvi

## Poglavje 5

### Zaključek

V diplomskem delu je predstavljena iBeacon tehnologija, njena uporaba pri razvoju aplikacij in podan je primer iBeacon aplikacije za platformi Android in iOS. Pri predstavitvi iBeacon tehnologije so predstavljene funkcionalnosti, ki jih tehnologija ponuja, razložen je pojem bližine, ki je osnoven koncept, ko govorimo o iBeaconih, ogledamo si alternativne tehnologije, na koncu je opravljen še podroben vpogled v konfiguracijo iBeaconov in vpliv konfiguracije na življenjsko dobo baterije. Pri konfiguraciji iBeaconov pridemo do ugotovitve, da povečanje intervala oddajanja močno podaljša življenjsko dobo baterije. Kljub temu je podano mnenje, da povečevanje intervala ni priporočljivo, saj še dodatno škodi že tako vprašljivi natančnosti določanja razdalje med napravo in iBeaconom.

V naslednjem delu, kjer je predstavljen razvoj aplikacij ob uporabi iBeacon tehnologije, razdelimo primere uporabe iBeaconov v tri skupine: določanje vstopa in izstopa iz regije, spremljanje razdalje do enega ali več iBeaconov in notranje lociranje. Nato sledi analiza dveh obstoječih aplikacij, ki uporabljata iBeacone, na koncu pa si ogledamo možnosti postavitve in vzdrževanja velikega števila iBeaconov. Tukaj pridemo do sklepa, da so možnosti nastavljanja parametrov in spremljanja stanja baterije velikega števila iBeaconov zelo omejene, kar je precejšna ovira za razvoj aplikacije, ki za delovanje potrebuje večje število iBeaconov. V zadnjem delu je predstavljen razvoj vzorčnih aplikacij za prikaz podatkov o pacientu za Android in iOS platformi, ki uporabljata iBeacon tehnologijo. Na

podlagi razvoja aplikacij za Android in iOS platformo je opravljena primerjava razvoja iBeacon aplikacije glede na platformo. V okviru te primerjave ugotovimo, da je razvoj na Android platformi v nekaterih elementih otežen, vendar je kljub temu povsem smiseln.

Na podlagi teoretične in empirične raziskave iBeacon tehnologije v diplomskem delu lahko na koncu pridemo do naslednjega sklepa. Najbolje je, da pri načrtovanju mobilnih aplikacij, ki uporabljajo iBeacon tehnologijo, upoštevamo, da je ta lahko precej nezanesljiva in da je mogoče relativno enostavne probleme (vstop/izstop iz regije in določanje najbližjega predmeta, kjer so predmeti razporejeni redko) rešiti učinkovito. Kompleksni problemi (določanje enega ali več najbližjih predmetov, kjer so predmeti razporejeni gosto in notranje lociranje) ponavadi zahtevajo neučinkovite rešitve (zelo veliko število iBeaconov, razvoj varovalnih mehanizmov v primeru, da iBeacon funkcionalnost odpove).

V prihodnosti bi bilo smiselno razširiti funkcionalnosti aplikacij. Smiselne razširitve funkcionalnosti: prikaz prvih nekaj najbližjih pacientov, izmed katerih uporabnik izbere tistega, ki ga zanima, implementacije prijave uporabnika, možnosti določanja najmanjše potrebne razdalje do pacienta, da je prepoznan kot najbližji in implementacija iskanja pacienta po imenu (za situacije, ko iBeacon funkcionalnost zataji). Z dodatnim testiranjem bi bilo mogoče razviti bolj zanesljiv algoritem za določanje najbližjega iBeacona.

# Literatura

- [1] Aislelabs. The hitchhikers guide to ibeacon hardware: A comprehensive report by aislelabs, 2015. [Spletni vir; uporabljen 24. avgusta 2015; dostopno na: <http://www.aislelabs.com/reports/beacon-guide/>].
- [2] Aislelabs. ibeacon range with specified tx power, 2015. [Spletni vir; uporabljen 24. avgusta 2015; dostopno na: <http://www.aislelabs.com/wp-content/uploads/2014/11/ibeacon-range-tx-power-meters.png>].
- [3] AltBeacon. Android beacon library, 2015. [Spletni vir; uporabljen 7. septembra 2015; dostopno na: <http://altbeacon.github.io/android-beacon-library/>].
- [4] Apple. Core location framework reference, 2015. [Spletni vir; uporabljen 7. septembra 2015; dostopno na: [https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation\\_Framework/](https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation_Framework/)].
- [5] Apple. Swift. a modern programming language that is safe, fast, and interactive, 2015. [Spletni vir; uporabljen 7. septembra 2015; dostopno na: <https://developer.apple.com/swift/>].
- [6] ARMbed. Understanding the different types of ble beacons, 2015. [Spletni vir; uporabljen 26. avgusta 2015; dostopno na: <https://developer.mbed.org/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/>].

- [7] Blackberry. Ble protocol stack, 2015. [Spletni vir; uporabljen 19. avgusta 2015; dostopno na: <https://supportforums.blackberry.com/t5/image/serverpage/image-id/16965i76FE4963F3BCCB84/image-size/original?v=mpbl-1&px=-1>].
- [8] Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml). *World Wide Web Consortium Recommendation REC-xml-19980210*. <http://www.w3.org/TR/1998/REC-xml-19980210>, 16, 1998.
- [9] Giorgio Conte, Massimo De Marchi, Alessandro A Nacci, Vincenzo Rana, and Donatella Sciuto. Bluesentinel: a first approach using ibeacon for an energy efficient occupancy detection system. In *1st ACM International Conference on Embedded Systems For Energy-Efficient Buildings (BuildSys)*, 2014.
- [10] Estimote. ibeacon region proximity zones, 2015. [Spletni vir; uporabljen 23. avgusta 2015; dostopno na: <http://www.aislelabs.com/reports/beacon-guide/>].
- [11] Matthew S Gast. *Building Applications with IBeacon: Proximity and Location Services with Bluetooth Low Energy*. "O'Reilly Media, Inc.", 2014.
- [12] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9):11734–11753, 2012.
- [13] James Gosling. *The Java language specification*. Addison-Wesley Professional, 2000.
- [14] Indoors. San francisco international airport, 2015. [Spletni vir; uporabljen 7. septembra 2015; dostopno na: <http://indoo.rs/sfo/>].
- [15] Kontakt.io. How to succeed in large beacon deployments?, 2015. [Spletni vir; uporabljen 7. septembra 2015; dostopno

- na: <http://gofor.kontakt.io/large-beacon-deployments-webinar/?category=lgeneration&from=resources&medium=cta>].
- [16] Kontakt.io. What is eddystone? 5 key facts about the new open beacon format from google, 2015. [Spletni vir; uporabljen 26. avgusta 2015; dostopno na: <http://kontakt.io/blog/what-is-eddystone/>].
- [17] Launch Here. Launch here ibeacon based app shortcuts, 2015. [Spletni vir; uporabljen 7. septembra 2015; dostopno na: <http://launchhere.awwapps.com>].
- [18] Vijay Madisetti and Arshdeep Bahga. *Internet of Things*. VPT, 2014.
- [19] Marand. Ehrscape - build amazing healthcare apps and services, 2015. [Spletni vir; uporabljen 7. septembra 2015; dostopno na: <https://www.ehrscape.com>].
- [20] Isaak D Mayergoyz. *Mathematical models of hysteresis and their applications*. Academic Press, 2003.
- [21] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.
- [22] Matt Neuburg. *iOS 7 Programming Fundamentals: Objective-c, xcode, and cocoa basics*. "O'Reilly Media, Inc.", 2013.
- [23] Nodes. List of the 9 biggest beacon manufacturers, 2015. [Spletni vir; uporabljen 2. septembra 2015; dostopno na: <http://www.nodesagency.com/list-9-biggest-beacon-manufacturers/>].
- [24] John Padgett, Karen Scarfone, and Lily Chen. Guide to bluetooth security. *NIST Special Publication*, 800(121):25, 2012.
- [25] Smart Insights. Mobile marketing statistics 2015, 2015. [Spletni vir; uporabljen 3. septembra 2015; dostopno na: [http:](http://)

[//www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/](http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/)].

- [26] Spring. Spring framework, 2015. [Spletni vir; uporabljen 7. septembra 2015; dostopno na: <http://projects.spring.io/spring-framework/>].
- [27] Pedro Teixeira. *Professional Node.js: Building Javascript based scalable software*. John Wiley & Sons, 2012.
- [28] The Evil Boss. Replacing the battery in an estimote ibeacon, 2015. [Spletni vir; uporabljen 30. avgusta 2015; dostopno na: <http://www.theevilboss.com/blog/2014/8/10/replacing-the-battery-in-an-estimote-ibeacon>].
- [29] VincentH Blog. Building cross-platform ibeacon apps for ios, android and windows with c and xamarin, 2015. [Spletni vir; uporabljen 4. septembra 2015; dostopno na: <http://vincenth.net/blog/archive/2014/04/24/building-cross-platform-ibeacon-apps-for-ios-android-and-windows-with-c-and-xamarin.aspx>].
- [30] Huang Xuguang. An introduction to android. *Database Lab. Inha University*, 2009.
- [31] Belén Cruz Zapata. *Android Studio Application Development*. Packt Publishing Ltd, 2013.